PREM: PREFIX-STEERED RECURRENT MEMORY FOR LONG-VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video understanding must capture transient visual evidence under strict token budgets, yet existing methods compress frames, append memory tokens, or alter internal key-value (KV) caches. We introduce **Prefix-Steered Recurrent Memory** (**PREM**), a memory-token-free framework for frozen vision-language models (VLMs). PREM separates video ingestion from query answering: a recurrent writer distills visual streams into a compact ≈ 256 KiB multi-slot associative state, while a question-conditioned readout adds memory-derived key/value (K/V) steering modulations to existing non-visual prompt prefixes during prefill. This enables write-once, query-many inference without extra prompt tokens or decoding recurrence. Across six long-video benchmarks, in offline and streaming end-of-stream settings, macro-average accuracy improves over frozen baselines at every evaluated decoder budget. Under a constrained budget of 16 frames, PREM improves macro-average accuracy by 3.06% on Qwen2.5-VL-3B, with gains of 11.0% on action antonym identification and 9.9% on localized needle retrieval. The added parameters are $\approx 0.24\%$ of the backbone parameter count, at 0.03 GiB of peak GPU memory overhead. Code is available at `https://anonymous.4open.science/r/PReM`.

1 INTRODUCTION

Long-video understanding requires multimodal models to reason over visual evidence scattered across extended temporal horizons (Wu et al., 2024; Fu et al., 2025; Mangalam et al., 2023). In practical domains such as continuous video streams, egocentric perception (Grauman et al., 2022), instructional procedures (Tang et al., 2019), and robotic manipulation and navigation (Brohan et al., 2023; Anwar et al., 2024), decisive evidence is often brief and surrounded by uninformative content. Effective reasoning thus demands both comprehensive temporal coverage and high sensitivity to subtle cues. However, modern vision-language models (VLMs) are constrained by strict visual token budgets due to quadratic attention complexity and GPU memory limits.

This tension creates an inevitable trade-off between temporal breadth and sequence cost. Aggressive frame downsampling risks discarding transient events entirely, while admitting more visual tokens incurs steep computational overhead and dilutes attention, frequently inducing "lost-in-the-middle" failures (Liu et al., 2024). As illustrated in Figure 1, existing approaches navigate this trade-off through three main paradigms: **(1) Visual context compression** reduces spatiotemporal redundancy via pooling, frame selection, or multi-rate sampling (Cheng et al., 2024; Shen et al., 2025; Li et al., 2026; Xu et al., 2024), yet information discarded during compression cannot be recovered for subsequent reasoning; **(2) Video memory banks** maintain episodic feature pools or explicit token banks (He et al., 2024; Song et al., 2024; Wang et al., 2025; Zhang et al., 2025; Diko et al., 2025; Liu et al., 2026), but their token-based readouts directly occupy prompt sequence capacity; and **(3) KV-cache management** dynamically prunes, merges, or evicts intermediate attention caches (Qin et al., 2025; Kim et al., 2025; Di et al., 2025; Ning et al., 2025; Yang et al., 2025; Chen et al., 2026), which avoids prompt token overhead but couples directly to internal attention history.

Consequently, historical visual evidence in existing paradigms either competes for prompt sequence budget or couples directly to internal attention caches. To address this challenge, we propose **Prefix-Steered Recurrent Memory (PREM)**, a dual-pathway framework that separates video ingestion from query answering. During a one-time ingestion pass, a query-agnostic recurrent writer distills

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.

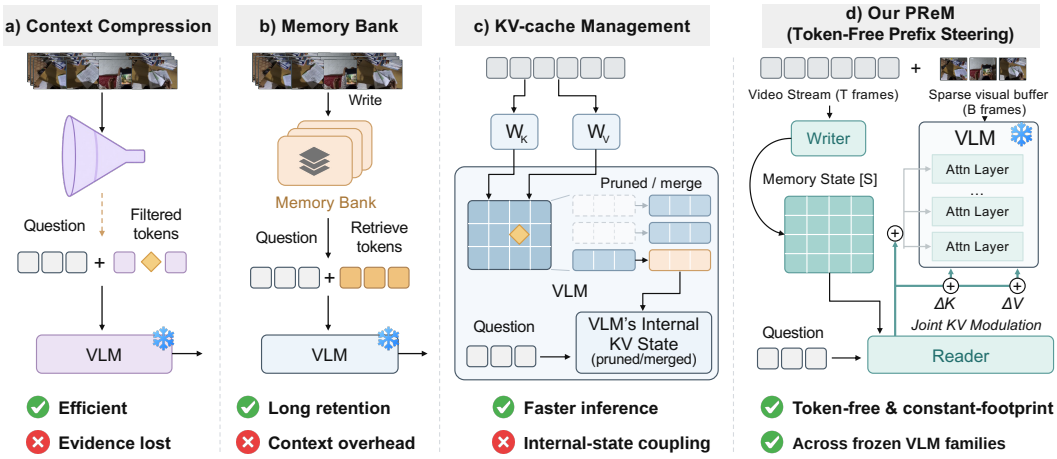

Figure 1: Comparison of long-video VLM paradigms. **(a)** Compression reduces visual context. **(b)** Memory banks consume prompt capacity. **(c)** KV management operates on internal attention states. **(d)** PREM steers the backbone via prefix modulations $(\Delta K, \Delta V)$ without sequence tokens.

incoming visual streams into a compact, multi-slot associative state (≈256 KiB) using selective gating to filter redundancy and protect salient cues. At query time, a question-conditioned readout retrieves memory content and injects it as additive key/value (K/V) steering modulations into non-visual prompt prefixes exclusively during prefill. Prefix key modulations re-weight attention allocated to the prefix, while value modulations inject long-horizon context to guide visual evidence utilization across subsequent layers. This design enables write-once, query-many inference over the visual buffer without appending prompt tokens, altering visual caches, or requiring recurrent decoding.

Our main contributions are summarized as follows:

- **Decoupled Memory Architecture.** We introduce a memory-token-free framework for frozen VLMs that ingests video streams into an external, constant-footprint recurrent state and reads that state at query time without consuming prompt sequence capacity.
- **Recurrent Memory with Prefill-Only Steering.** We design a multi-factor gated write update to selectively filter temporal redundancy, coupled with a question-conditioned readout that steers prefix K/V representations during prefill, eliminating recurrent updates during generation.
- **Comprehensive Empirical Validation.** Across six long-video benchmarks and three VLM families, PREM improves macro-average accuracy at every evaluated decoder budget in both offline and streaming end-of-stream settings. Under a constrained budget of 16 frames, it achieves a 3.06% macro-average gain on Qwen2.5-VL-3B (with task-specific improvements up to 11.0%) while adding ≈0.24% of the backbone parameter count, at 0.03 GiB peak GPU memory overhead.

2 Related Work

Visual context compression. Visual context compression alleviates quadratic attention costs by pruning tokens along different dimensions. VideoLLaMA 2 (Cheng et al., 2024) aggregates spatiotemporal features via convolutional connectors, while LLaMA-VID (Li et al., 2024b) distills each frame into dual context-content tokens. To balance spatial detail and motion, SlowFast-LLaVA (Xu et al., 2024) adopts asymmetric frame-rate pathways. Complementarily, LongVU (Shen et al., 2025) and VideoChat-Flash (Li et al., 2026) employ text-guided frame selection and hierarchical token compression, respectively, whereas LongVILA (Chen et al., 2024) scales context via multimodal sequence parallelism. However, information discarded during compression cannot be recovered for subsequent generation. In contrast, PREM maintains a compact visual buffer for direct grounding while accumulating extended temporal context into an external recurrent state.

Video memory banks. To maintain historical context beyond the immediate window, token-based memory banks accumulate and cluster visual features across extended horizons. MA-LMM (He et al., 2024) accumulates online features in memory banks, while MovieChat (Song et al., 2024) separates tokens into short- and long-term stores inspired by the Atkinson–Shiffrin model. VideoLLaMB

(Wang et al., 2025) propagates recurrent memory tokens across segmented scenes, while ReWind (Diko et al., 2025) leverages instruction-conditioned memory to retrieve salient frames for reasoning. To preserve streaming details, Flash-VStream (Zhang et al., 2025) and VideoChat-Online (Huang et al., 2025) incorporate multi-resolution augmentation buffers. However, these methods expose memory by appending auxiliary tokens directly to the prompt or relying on backbone-coupled bridges, which exhaust sequence capacity. PREM avoids sequence overhead by keeping its recurrent state outside the prompt and accessing it via additive prefix steering.

KV-cache management. Managing intermediate attention key-value (KV) states offers a token-free alternative. Video-XL-2 (Qin et al., 2025) and InfiniPot-V (Kim et al., 2025) compress caches via task-aware sparsification or value-norm pruning. For long-term retrieval, ReKV (Di et al., 2025) offloads chunked KV states to host memory, while StreamMem (Yang et al., 2025) aggregates incoming caches into query-agnostic fixed buffers. While effective, these methods operate directly on high-dimensional model-internal activations, tying memory management to backbone-specific attention representations and cache formats. PREM constructs an external recurrent state directly from visual representations and modulates prefix K/V states exclusively during prefill.

Fast weights and associative memory. Fast-weight models and linear Transformers maintain context within dynamically updated parameters (Schmidhuber, 1992; Katharopoulos et al., 2020; Schlag et al., 2021; Ramsauer et al., 2021). Recent systems adapt this for long sequences: Titans (Behrouz et al., 2025) and StreamTTT (Chen et al., 2026) employ test-time training to optimize parameters online, introducing non-trivial latency. Closest in interface, δ-mem (Lei et al., 2026) generates low-rank attention corrections during autoregressive generation. PREM departs from these designs by employing closed-form, multi-factor gated outer-product writes to bypass online optimization, and applying modulations exclusively during prefill to eliminate recurrent computation during generation.

3 PREFIX-STEERED RECURRENT MEMORY

Problem formulation and overview. Given an extended video $\mathcal{V} = \{v_t\}_{t=1}^{T}$ of T frames and a textual query $\mathcal{Q}$, long-video understanding aims to generate an answer A conditioned on both modalities. PREM addresses this by decoupling temporal retention from sequence capacity via a dual-pathway interface (Figure 2). The primary pathway feeds a bounded visual buffer $\mathcal{V}_B \subset \mathcal{V}$ containing B frames into a frozen VLM backbone for visual grounding. Concurrently, a recurrent writer ingests the frame sequence from visual encoder representations, updating an external, constant-footprint multi-slot associative state S while bypassing the LLM backbone during ingestion. At query time, a question-conditioned readout projects S into steering vectors $d_i^{k,(\ell)}, d_i^{v,(\ell)}$ and adds them only to non-visual prefix tokens during prefill. Prefix keys re-weight attention allocated to the prefix, while prefix values inject the retrieved state across layers, without extra tokens or backbone fine-tuning.

3.1 QUERY-AGNOSTIC RECURRENT WRITE

Let $E(v_t)_p \in \mathbb{R}^{d_{\text{enc}}}$ denote the feature from the frozen vision encoder for frame $v_t \in \mathcal{V}$ at spatial patch $p \in \{1, \ldots, P_t\}$, where P_t is the number of patch tokens at step t. We obtain the frame-level visual feature x_t through spatial mean-pooling and project it into address and content spaces:

$$x_t = \frac{1}{P_t} \sum_{p=1}^{P_t} E(v_t)_p, \qquad k_t = W_k \,\mathrm{LN}(x_t), \quad u_t = W_v \,\mathrm{LN}(x_t), \tag{1}$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization. The learnable matrices $W_k \in \mathbb{R}^{d_k \times d_{\text{enc}}}$ and $W_v \in \mathbb{R}^{d_v \times d_{\text{enc}}}$ project x_t into address and content spaces, respectively. Writer features follow the visual-to-text projection ($d_{\text{enc}} = d_{\text{model}}$); spatial patch tokens remain available in the visual buffer.

Given the sequence $\{x_t\}_{t=1}^{T}$, the recurrent state maintains K matrix slots $S_m \in \mathbb{R}^{d_v \times d_k}$ and slot confidences $c_m \in [0, 1]$, where d_k and d_v denote key and value dimensions. A learnable address transform $A_m \in \mathbb{R}^{d_k \times d_k}$, initialized near identity, parameterizes each slot:

$$\kappa_{t,m} = \mathrm{norm}(A_m k_t), \qquad e_{t,m} = u_t - S_m \kappa_{t,m}, \tag{2}$$

where $\mathrm{norm}(v) = v / \max(\|v\|_2, \epsilon_{\text{norm}})$ denotes ℓ_2 normalization with numerical stabilizer $\epsilon_{\text{norm}} = 10^{-12}$. The vector $\kappa_{t,m} \in \mathbb{R}^{d_k}$ is the unit-norm write address, and residual $e_{t,m} \in \mathbb{R}^{d_v}$ measures the discrepancy between projected value u_t and slot prediction $S_m \kappa_{t,m}$.

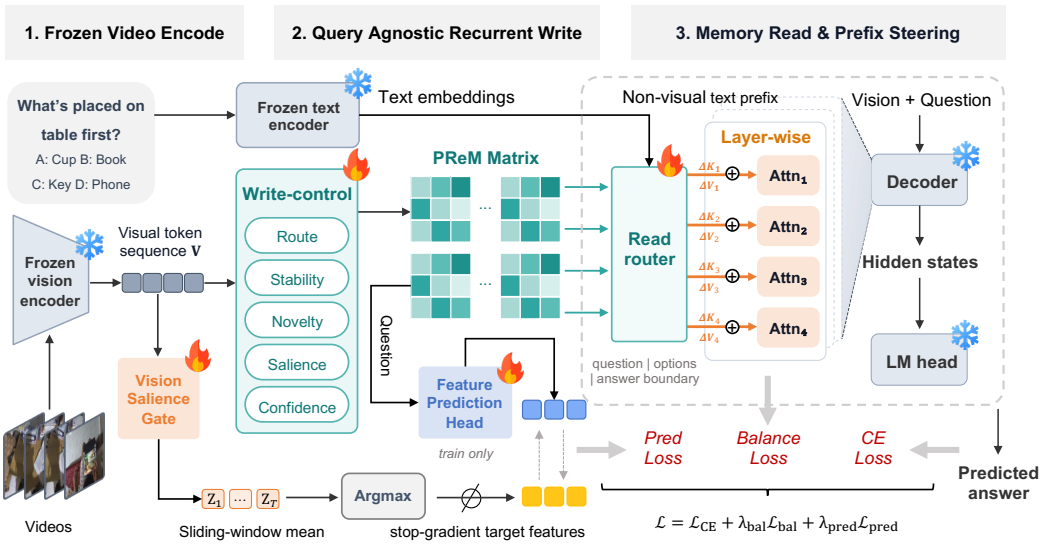

Figure 2: PREM architecture and supervision. A query-agnostic gated residual rule updates a K-slot recurrent state S. During QA, prefix representations route memory reads into additive K/V steering. The visual buffer remains standard input; state S operates outside the prompt with zero tokens.

As illustrated in Figure 3, five bounded gates regulate memory updates over extended horizons:

$$\begin{aligned} \pi_t^w &= \mathrm{softmax}(r_w k_t), & b_{t,m} &= \sigma((W_b k_t)_m), \\ n_{t,m} &= \mathrm{clip}\left(\frac{\|e_{t,m}\|_2}{\max(\|u_t\|_2, \epsilon)}, 0, 1\right), & z_t &= \sigma(\phi(\mathrm{LN}(x_t))), \\ a_{t,m} &= [1 - c_m(1 - z_t)]_+, & g_{t,m} &= \pi_{t,m}^w b_{t,m} n_{t,m} z_t a_{t,m}, \end{aligned} \tag{3}$$

where $\sigma(\cdot)$ is sigmoid, $[\cdot]_+ = \max(0, \cdot)$ is ReLU, and $\epsilon = 10^{-6}$ is a numerical stabilizer. Affine gate biases are implicit. The five gates regulate updates as follows: **(1) Slot routing** ($\pi_t^w \in \Delta^K$) maps k_t via $r_w \in \mathbb{R}^{K \times d_k}$ for semantic slot specialization; **(2) Update stability** ($b_{t,m} \in (0,1)$) uses $W_b \in \mathbb{R}^{K \times d_k}$ to scale steps and suppress perturbations; **(3) Residual novelty** ($n_{t,m} \in [0,1]$) suppresses redundant writes when $e_{t,m} \approx 0$; **(4) Visual salience** ($z_t \in (0,1)$) downweights background frames via ϕ, a $d_{\text{model}} \to \lfloor d_{\text{model}}/4 \rfloor \to 1$ GELU MLP; and **(5) Confidence protection** ($a_{t,m} \in [0,1]$) protects confident slots ($c_m \to 1$) from low-salience writes ($z_t \to 0$).

The gate $g_{t,m}$ modulates the outer-product update for each slot matrix S_m and its confidence c_m:

$$S_m \leftarrow \sigma(f_m) S_m + \eta_w g_{t,m} e_{t,m} \kappa_{t,m}^\top, \qquad c_m \leftarrow \max\{\gamma_c c_m, \mathrm{sg}(g_{t,m} b_{t,m})\}, \tag{4}$$

where $f_m \in \mathbb{R}$ is a learnable slot-wise forget logit, and $\sigma(f_m)$ controls gradual memory decay. The fixed factor $\eta_w > 0$ sets the write step size, while $\gamma_c \in (0,1)$ sets the exponential confidence decay. The stop-gradient operator $\mathrm{sg}(\cdot)$ blocks confidence gradients. The recurrent state occupies $\mathcal{O}(K d_v d_k)$ memory (≈256 KiB for K=4 and d_v=d_k=128) at any video length, with one write per frame.

3.2 Question-Conditioned Read and Prefix Steering

Because the recurrent write update is query-agnostic (Eqs. 1–4), the multi-slot state $S = \{S_m\}_{m=1}^K$ supports repeated queries after a single video ingestion pass. At query time, we average embeddings of existing non-visual prefix tokens (instructions, query $\mathcal{Q}$, options, and prompt boundaries) into a global question representation $q \in \mathbb{R}^{d_{\text{model}}}$. To accommodate different abstraction levels across depth, we partition the L backbone layers into G contiguous groups, where layer $\ell \in \{1, \ldots, L\}$ maps to group $g(\ell) = \lfloor (\ell - 1)G/L \rfloor \in \{0, \ldots, G-1\}$.

At layer ℓ and prefix position i, the hidden state $h_i^{(\ell)} \in \mathbb{R}^{d_{\text{model}}}$ is conditioned on q and projected to the normalized read address $\xi_{i,m}^{(\ell)} \in \mathbb{R}^{d_k}$, retrieving value $r_{i,m}^{(\ell)} \in \mathbb{R}^{d_v}$ from slot m:

$$\begin{aligned} \tilde{h}_i^{(\ell)} &= \mathrm{LN}(h_i^{(\ell)}) + W_c \,\mathrm{LN}(q), & q_i^{(\ell)} &= W_r \,\mathrm{LN}(\tilde{h}_i^{(\ell)}), \\ \xi_{i,m}^{(\ell)} &= \mathrm{norm}(A_m q_i^{(\ell)}), & r_{i,m}^{(\ell)} &= S_m \xi_{i,m}^{(\ell)}, \end{aligned} \tag{5}$$

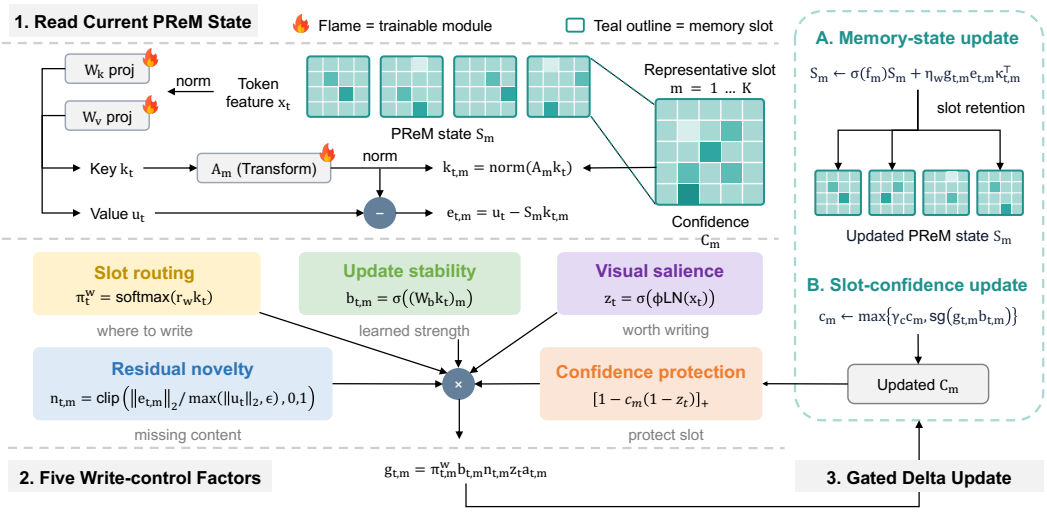

Figure 3: Recurrent write update. Subtracting slot predictions from visual projections yields associative residuals, modulated by multi-factor controls to update decayed memory and slot confidences.

where $W_c \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ is a conditioning projection, $W_r \in \mathbb{R}^{d_k \times d_{\text{model}}}$ maps conditioned features to query space, and A_m is the slot address transform from Eq. 2.

A router $R : \mathbb{R}^{3d_v} \to \mathbb{R}$ evaluates conditioned query $q_i^{(\ell)}$, retrieved value $r_{i,m}^{(\ell)}$, and their Hadamard product, producing relevance logits $s_{i,m}^{(\ell)}$ that normalize into routing weights $\rho_{i,:}^{r,(\ell)} \in \Delta^K$:

$$s_{i,m}^{(\ell)} = R([W_{qv} q_i^{(\ell)}; r_{i,m}^{(\ell)}; W_{qv} q_i^{(\ell)} \odot r_{i,m}^{(\ell)}]), \qquad \rho_{i,:}^{r,(\ell)} = \text{softmax}_m(s_{i,m}^{(\ell)}/\tau), \tag{6}$$

where $[\cdot;\cdot]$ denotes concatenation, $\odot$ the Hadamard product, and $\tau > 0$ the routing temperature. $W_{qv} \in \mathbb{R}^{d_v \times d_k}$ aligns dimensions; R is a two-layer GELU router (Table 4). Group-specific heads $H_{g(\ell),m}^a \in \mathbb{R}^{d_{\text{model}} \times d_v}$ ($a \in \{k, v\}$) map weighted reads to steering vectors $d_i^{a,(\ell)} \in \mathbb{R}^{d_{\text{model}}}$:

$$d_i^{a,(\ell)} = \frac{\alpha_{\text{eff}}}{K} \sum_{m=1}^{K} \rho_{i,m}^{r,(\ell)} H_{g(\ell),m}^a r_{i,m}^{(\ell)}, \qquad \alpha_{\text{eff}} = \exp(\hat{\alpha}) \alpha_{\text{run}}, \tag{7}$$

where $\hat{\alpha} \in \mathbb{R}$ is a learnable scale logit and α_{run} is an inference-time scale. The steering vectors define a hidden-state perturbation before the frozen K/V projections:

$$K_i^{(\ell)} = W_K^{(\ell)}(h_i^{(\ell)} + d_i^{k,(\ell)}), \qquad V_i^{(\ell)} = W_V^{(\ell)}(h_i^{(\ell)} + d_i^{v,(\ell)}), \tag{8}$$

where $W_K^{(\ell)}, W_V^{(\ell)}$ are frozen projections (biases omitted). The perturbations induce additive shifts $\Delta K = W_K d^k$ and $\Delta V = W_V d^v$ at the projection outputs, followed by the backbone's native attention transforms (Appendix A.5).

Prefill-only steering. Modulation applies to non-visual prompt prefixes during prefill, including instructions and the question $\mathcal{Q}$. Prefix keys (ΔK) modulate the attention received by prefix positions, while prefix values (ΔV) inject the retrieved memory representations into subsequent layers. After prefill, the steered prefix entries stay in the standard KV cache, and generation reuses them. Appendices A.5 and A.6 detail the attention interface and initialization.

3.3 Training and Inference Protocols

During training, videos are sampled uniformly at a fixed frame rate (capped at $T \leq 240$ frames for long videos; Appendix A.8). The writer ingests an incoming sequence of T frames to update the recurrent state S, while the decoder prompt draws a visual buffer $\mathcal{V}_B$ of B frames from the same stream. We optimize solely PREM parameters by minimizing:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{bal}} \mathcal{L}_{\text{bal}} + \lambda_{\text{pred}} \mathcal{L}_{\text{pred}}, \tag{9}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, and $\lambda_{\text{bal}}, \lambda_{\text{pred}}$ weight auxiliary objectives. The load-balancing term $\mathcal{L}_{\text{bal}} = \sum_m (\bar{\rho}_m - 1/K)^2$ averages read-routing weights $\bar{\rho}_m$ over valid prefixes. The prediction loss masks a high-salience temporal span during a second pass and reconstructs its normalized value features from the held-out state, conditioned on the question and temporal position. Appendix A.7 details the head, shapes, target, and loss. During inference, PREM applies the same update and modulation interfaces (Eqs. 4 and 8) in offline and streaming end-of-stream modes: offline ingestion processes the full stream once, whereas streaming carries state (S, c) across chunks and answers after the video ends, both predicting A via one-token constrained decoding.

4 EXPERIMENTS

We evaluate PREM across four questions:

- **RQ1 (Performance):** How does PREM compare with long-video and streaming methods? (§4.2)
- **RQ2 (Scaling):** Do the gains hold across decoder budgets, durations, and backbones? (§4.3)
- **RQ3 (Ablations):** What are the contributions of prefix steering and write gating, and how stable is the design under training horizon and hyperparameters? (§4.4)
- **RQ4 (Efficiency):** Where do efficiency and task gains originate under matched settings? (§4.5)

4.1 EXPERIMENTAL SETUP

Models and baselines. We instantiate PREM on three frozen backbones: Qwen2.5-VL-3B, Qwen3-VL-8B, and LLaVA-Video-7B (Bai et al., 2025b;a; Zhang et al., 2024). Per Table 1, baselines include: (1) **specialized architectures** spanning compression (LLaMA-VID, VideoLLaMA2, LLaVA-Video, LongVILA, LongVU) and streaming memory (MovieChat, StreamChat, ReKV, Flash-VStream) under native budgets; and (2) **generalist base VLMs** (Qwen2.5-VL-3B, Qwen3-VL-8B). We train only PREM for one epoch on a LLaVA-Video-178K subset (6,430 videos, 17,849 QA pairs, $< 1.4\%$ of its QA pairs; Table 3). Trainable parameters total 9.07M (Qwen2.5) and 27.48M (Qwen3) with $d_v{=}d_k{=}128$ ($\approx$0.24%–0.30%). The standard configuration uses $K{=}4$; Table 2 uses $K{=}1$ for the controlled comparison. Training samples 1 FPS with write cap $T{=}64$ and buffer B.

Evaluation protocol. Evaluation spans six benchmarks: LongVideoBench (LVB) (Wu et al., 2024), MLVU (Zhou et al., 2025), Video-MME (VMME) (Fu et al., 2025), EgoSchema (EgoS) (Mangalam et al., 2023), MVBench (MVB) (Li et al., 2024a), and LVBench (LVBN) (Wang et al., 2024). Models predict via one-token constrained decoding; we report per-benchmark accuracy and macro **Avg**. Per §3.3, evaluation ingests a question-agnostic 1 FPS stream, uniformly subsampling any video longer than 240 s to at most $T{=}240$ write frames. Matched controls fix the backbone, prompt template, decoder budget $B \in \{16, 32, 64, 90\}$, resolution cap, and decoding rule; Base ($B{=}240$) is a high-budget reference. Offline mode writes that stream once. Streaming end-of-stream mode carries state (S, c) across two-frame chunks and answers after the video ends (Appendix A).

4.2 RQ1: PERFORMANCE COMPARISON

Table 1 provides system-level context from specialized long-video systems at native budgets and measures gains against frozen base VLMs at the same decoder budget ($B{=}90$). PREM remains competitive with specialized long-video systems under their native configurations. Across both backbones, macro-average accuracy rises over the paired frozen bases: offline Avg by $+1.89\%$ (3B) and $+1.00\%$ (8B), and streaming Avg by $+1.09\%$ (3B) and $+1.43\%$ (8B). In streaming, PREM (Qwen3-VL-8B) reaches 64.72% Avg while training only 27.5M parameters ($< 0.30\%$).

4.3 RQ2: BUDGET SCALING AND GENERALIZATION

Budget scaling and evaluation across backbones. Figure 4 shows a macro-average gain at every $B \in \{16, 32, 64, 90\}$, largest when the visual buffer is sparsest. On Qwen2.5-VL-3B at $B{=}16$, offline Avg rises by **+3.06%** (50.39% to 53.45%) and streaming Avg by **+1.45%**. Three training seeds put that offline gain at $+2.99 \pm 0.06\%$, well beyond the seed variance (Table 11). Qwen3-VL-8B peaks at **+2.00%** offline ($B{=}64$) and **+1.58%** streaming ($B{=}16$); its offline $B{=}16$ seeds give $+0.95 \pm 0.03\%$, confirming stability across runs. LLaVA-Video-7B margins are $+0.20$–$+1.29\%$ (Table 6).

Table 1: **Public-benchmark comparison** (%). External systems follow official configurations (Appendix A). Base VLMs and PREM share decoder budget B=90. Subscripts denote paired gains over Base. Bold marks top performance within each regime.

Method (Backbone)	Tuning / Budget	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg
Offline Ingestion *(Specialized long-video architectures)*								
LLaMA-VID (Vicuna-7B)	Full-FT (1 FPS)	28.27	35.26	30.11	27.20	32.78	25.89	29.92
VideoLLaMA2 (Qwen2-7B)	Full-FT (B=16)	48.54	56.92	54.59	61.20	55.43	36.73	52.24
LLaVA-Video (Qwen2-7B)	SFT (B=64)	59.39	70.02	63.33	58.20	59.60	42.41	58.82
LongVILA (Qwen2-7B)	Long-FT (B=32/256)	55.27	65.33	60.11	72.20	64.65	40.99	59.76
LongVU (Qwen2-7B)	SFT (0.5 FPS)	52.80	65.06	56.30	70.00	64.43	37.70	57.72
StreamChat (LongVA-7B)	Train-free (20%)	46.00	54.30	46.15	47.20	45.73	35.31	45.78
ReKV (LLaVA-OV-7B)	Train-free (0.5 FPS)	56.17	67.03	57.67	60.80	55.43	39.77	56.15
Flash-VStream (Qwen2-VL-7B)	LoRA (B=240)	52.06	62.71	58.19	69.60	60.42	38.22	56.87
Generalist base VLMs & PREM (decoder B=90)								
Qwen2.5-VL-3B	Frozen (B=90)	52.13	63.45	61.26	61.20	57.92	40.41	56.06
PREM (Qwen2.5-VL-3B)	Plug-in (B=90)	$55.27_{+3.14}$	$65.47_{+2.02}$	$61.30_{+0.04}$	61.20	$61.35_{+3.43}$	$43.12_{+2.71}$	$57.95_{+1.89}$
Qwen3-VL-8B	Frozen (B=90)	60.88	70.94	67.56	72.60	65.90	45.32	63.87
PREM (Qwen3-VL-8B)	Plug-in (B=90)	$\mathbf{63.05}_{+2.17}$	$\mathbf{71.22}_{+0.28}$	$\mathbf{67.87}_{+0.31}$	$\mathbf{74.20}_{+1.60}$	$\mathbf{67.10}_{+1.20}$	$\mathbf{45.76}_{+0.44}$	$\mathbf{64.87}_{+1.00}$
Streaming End-of-Stream *(Chronological Ingestion, native budgets)*								
MovieChat (Vicuna-7B)	Train-free (128×8)	21.17	28.55	26.44	20.40	29.38	23.56	24.92
StreamChat (LongVA-7B)	Train-free (20%)	46.00	53.20	46.63	49.00	45.68	34.41	45.82
ReKV (LLaVA-OV-7B)	Train-free (0.5 FPS)	56.17	67.03	57.67	60.80	55.43	39.83	56.16
Flash-VStream (Qwen2-VL-7B)	LoRA (B=240)	52.28	60.55	55.89	70.40	57.62	37.19	55.66
Generalist base VLMs & PREM (decoder B=90)								
Qwen2.5-VL-3B	Frozen (B=90)	54.75	63.59	59.63	61.40	61.20	41.77	57.06
PREM (Qwen2.5-VL-3B)	Plug-in (B=90)	$56.54_{+1.79}$	$65.52_{+1.93}$	$61.30_{+1.67}$	$61.20_{-0.20}$	$61.40_{+0.20}$	$42.93_{+1.16}$	$58.15_{+1.09}$
Qwen3-VL-8B	Frozen (B=90)	61.71	69.75	67.22	72.40	66.05	42.61	63.29
PREM (Qwen3-VL-8B)	Plug-in (B=90)	$\mathbf{62.15}_{+0.44}$	$\mathbf{71.22}_{+1.47}$	$\mathbf{67.87}_{+0.65}$	$\mathbf{74.20}_{+1.80}$	$\mathbf{67.13}_{+1.08}$	$\mathbf{45.76}_{+3.15}$	$\mathbf{64.72}_{+1.43}$

Generalization across native video durations. Figure 5 stratifies the four-slot diagnostic run by native video duration. On Video-MME, gains increase monotonically with video length: +0.67% on Short, +1.78% on Medium, and +1.89% on Long videos. On LVBench, the advantage widens to **+5.61%** on videos of at least 60 minutes, with the 95% bootstrap CI strictly above zero. This diagnostic run shows its largest gain in the longest LVBench group.

4.4 RQ3: Ablations and Training Robustness

Diagnostic ablations. Figure 6 reports component ablations on Qwen2.5-VL-3B (B=16) under a shared experimental setup. Full PREM reaches 53.45% Avg; K-only and V-only variants reach 51.66% and 51.92%. Removing individual write-control factors consistently reduces the aggregate score, with drops of 0.79–1.42 points relative to Full PREM. The zero-buffer probe (B=0) reaches 38.33%, +3.51 above the vision-free Base.

Training efficiency and horizon generalization. Figure 9a–b sweeps writer cap T against full-horizon evaluation (T=240, B=16). Even at T=16, enabling S boosts offline Avg by +2.17 over the clean Base, reaching 52.56%. The standard T=64 configuration reaches 53.45% offline and 53.46% streaming at B=16 (Table 5), while using 73.3% fewer writer steps than T=240.

Sensitivity. We evaluate 36 configurations and a clean Base reference across eight hyperparameters (Table 10; Figure 11). Training and capacity sweeps yield 51.29%–52.76% Avg. In the slot sweep, K=4 yields 52.65% Avg. All reported benchmarks use one shared configuration per backbone. Infer-

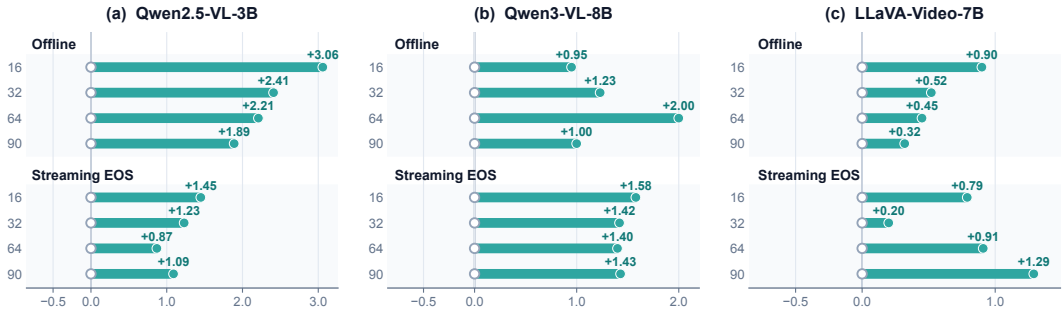

Figure 4: Horizontal axis: PREM − Base Avg gain; vertical axis: decoder budget $B \in \{16, 32, 64, 90\}$. Full breakdowns appear in Appendix Tables 5 and 6.

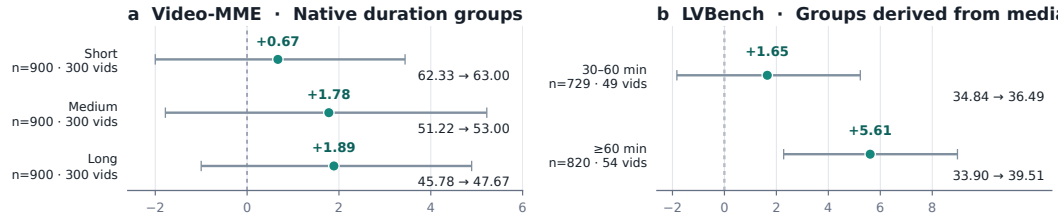

Figure 5: Duration-stratified PREM − Base gains on Qwen2.5-VL-3B (B=16, K=4, T_{mem}=16; Appendix B.1). Error bars: 95% paired bootstrap CIs (20k); labels: Base → PREM accuracy.

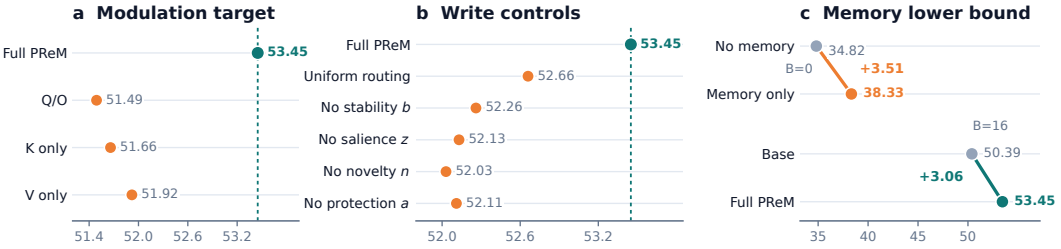

Figure 6: Diagnostic ablations on Qwen2.5-VL-3B (offline B=16). Horizontal axis: Avg accuracy (%). **(a)** Steering target interfaces. **(b)** Write gating terms against Full PREM (dashed line). **(c)** Zero-buffer probe (B=0): the state against a vision-free Base, with the B=16 Base as reference.

ence scales α_{run}=0.5/1.0 yield 52.26%/51.74%; larger scales 1.5/2.0 reduce Avg to 48.79%/43.56%. Multi-seed results appear in Appendix B.2.

4.5 RQ4: Efficiency and Task Analysis

Controlled streaming and task advantages. Table 2 compares three interfaces on Qwen2.5-VL-3B (B=16, T=240; Appendix A.4). Base and LoRA read the B-frame buffer (52.01%, 51.21%). Token-Readout runs the same writer on that stream and inserts ten prompt tokens (51.33%); InfiniPot-V writes the stream into its KV cache (51.73%, 84.24 MiB, 17.44 s). Parameter-matched prefix readout reaches 53.46%, +2.13 over that token readout and +1.73 over that KV cache, at 0.063 MiB with no added tokens. The shared stream does not by itself produce this margin. Task gaps are uneven (Figure 7): action antonym +11.0% and state change +9.0% on MVBench, needle retrieval +9.9% and count +9.2% on MLVU, against summarization −1.7% and order/topic reasoning −1.5% (Appendix Figure 10). PREM retains localized dynamic cues rather than acting as a temporal average.

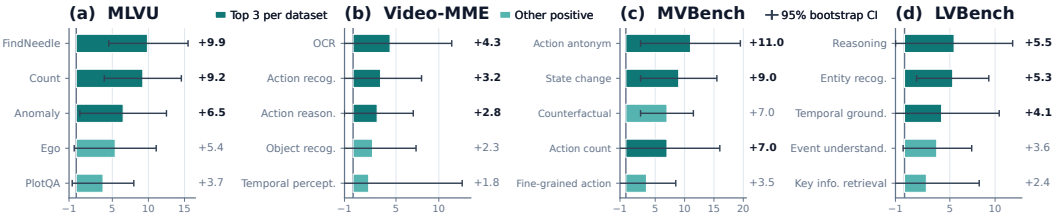

Figure 7: Highest-gain categories on Qwen2.5-VL-3B (offline B=16). Paired PREM − Base difference (%) with 95% CIs. Full breakdown in Appendix Figure 10.

Single-video case study. Figure 8 examines an EgoSchema question about banner text at 139 s. Ablating S_3 or S_4 reduces the gold-answer margin by 0.75; ablating S_1 or S_2 leaves it unchanged. Prefix steering changes the prediction from D to B. Slot-removal interventions indicate that specific slots contribute to decisions involving localized transient evidence.

Runtime efficiency. Figure 9c–d and Table 7 separate ingestion from answering. Peak GPU memory rises by 0.03 GiB at every budget. At B=16, mean latency rises from 5.94 s to 8.19 s because ingestion rises from 5.64 s to 7.81 s, while answering rises from 0.30 s to 0.38 s. The state (S, c) is

Table 2: **Controlled streaming comparison on Qwen2.5-VL-3B.** Matched streams (T=240, B=16) and parameter budgets. Bold and underline mark top and second-best results.

Method	Params (M)	Benchmark Accuracy (%)							Resource Accounting			
		LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg	State (MiB / Tok.)	VRAM (GiB)	Ingest (s)	Answer (s)
Base (Frozen, B=16)	0	<u>52.73</u>	55.68	53.89	<u>54.40</u>	59.98	35.38	<u>52.01</u>	0.000 / 0	7.43	5.64	0.30
+ Parameter Fine-tuning (LoRA, r=20)	9.22	49.36	55.45	53.04	**54.60**	57.98	<u>36.86</u>	51.21	0.000 / 0	7.46	5.64	0.30
+ Token-Readout (Memory Bank)	9.07	51.23	55.59	53.48	52.80	58.70	36.15	51.33	0.063 / +10	7.48	9.20	0.35
+ InfiniPot-V (KV-Cache Management)	0	52.06	<u>57.10</u>	**54.48**	50.20	**61.08**	35.44	51.73	84.236 / 0	7.99	17.44	0.32
+ PREM (Prefix Recurrent Memory)	9.07	**53.55**	**59.68**	<u>54.11</u>	**54.60**	<u>60.33</u>	**38.48**	**53.46**	0.063 / 0	7.46	7.81	0.38

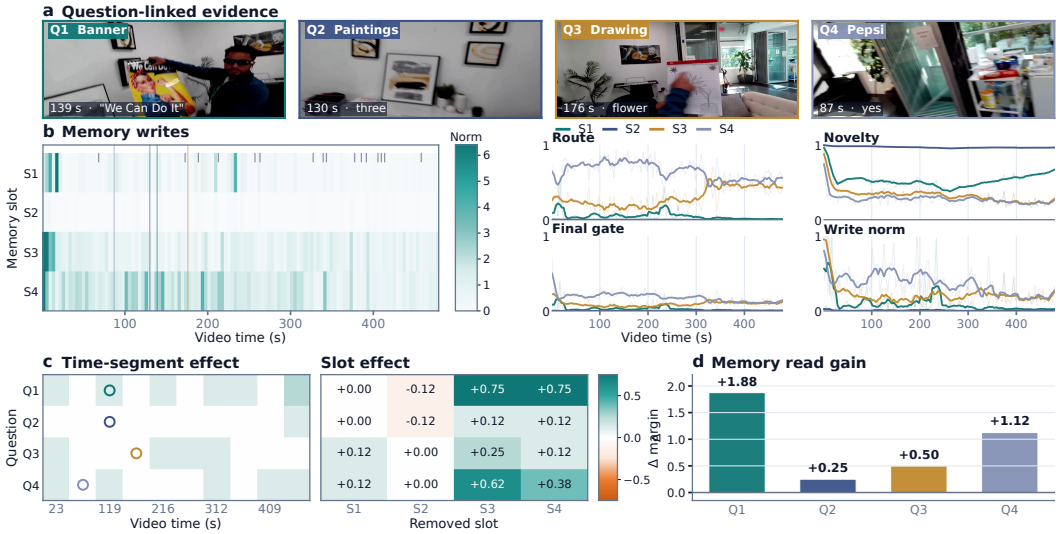

Figure 8: Qualitative write-read dynamics on EgoSchema. **(a)** Evidence frames. **(b)** Write intensity and gating. **(c)** Margin drop $m_{\text{full}} - m_{\text{ablated}}$ under temporal/slot removal, where m is the gold-answer logit minus the largest alternative logit. **(d)** Readout gain $m_{\text{on}} - m_{\text{off}}$.

built once per video. Under the measured mix, 41.4% of questions reuse it, and the 2.25 s mean gap is the average of that mix. From the second question on the same video, the added latency is the answer gap: 0.08 s at B=16, and within 0.24 s through B=90.

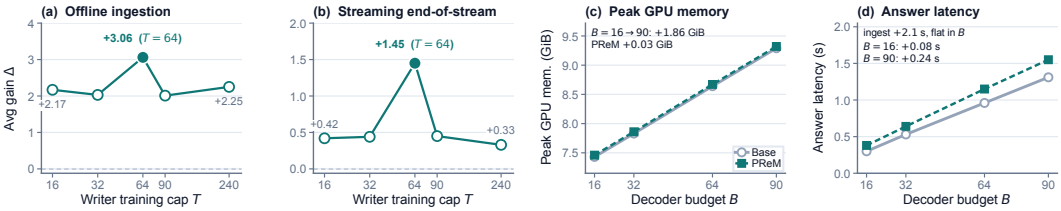

Figure 9: Training horizon and cost on Qwen2.5-VL-3B. **(a–b)** Writer cap T (T=240, B=16; Table 9); dashed: zero gain; filled: T=64. **(c)** Peak memory: B (16→90) costs +1.86 GiB, PREM +0.03 GiB. **(d)** A repeated query stays within 0.24 s (Table 7).

5 CONCLUSION

We presented PREM, a memory-token-free framework that equips frozen VLMs with long-horizon retention via prefix-steered recurrent memory. Query-agnostic ingestion and question-conditioned prefix steering keep a compact state without prompt tokens or cache edits. Gains are larger under stricter decoder budgets and on longer videos, with few added parameters. On a shared stream, prefix steering outperforms token and KV-cache readouts; task analyses show larger gains on localized transient evidence. The state is built once per video, so later queries add negligible latency. Future work adds grounded memory for embodied perception and planning.

AI USE STATEMENT

Generative AI tools assisted with manuscript critique, methodology and experiment feedback, language editing, LaTeX grammar checking, and text formatting. The authors independently made all methodological, implementation, experimental, and analytical decisions. All AI-assisted text and empirical claims were reviewed against source code, logs, and checkpoints, and the authors take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

The main text and appendix provide all details needed to reproduce the reported findings. Section 3 specifies the architecture, read/write interfaces, objectives, and ingestion protocols; Section 4.1 details training configurations, backbones, budgets, and metrics. Further implementation details, sensitivity sweeps, and multi-seed runs appear in Appendix A–B. Code is available at `https://anonymous.4open.science/r/PReM`.

ETHICS STATEMENT

This work studies long-video understanding on public academic benchmarks, training lightweight PREM modules on frozen VLMs without collecting human-participant data or private video assets. Dual-use applications such as surveillance fall outside our scope. Like its backbones, PREM may inherit web-scale biases; deployment should apply appropriate moderation and fairness safeguards. All authors adhere to the ICLR Code of Ethics.

REFERENCES

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. *arXiv preprint arXiv:2409.13682*, 2024.

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

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

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted

Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Joya Chen, Zeyun Zhong, and Mike Zheng Shou. Streamttt: Reconciling real-time perception and long-term memory in streaming vlms. *arXiv preprint arXiv:2608.13416*, 2026.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. LongVILA: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-LLMs. *arXiv preprint arXiv:2406.07476*, 2024.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Tao Zhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video KV-cache retrieval. In *International Conference on Learning Representations*, 2025.

Anxhelo Diko, Tinghuai Wang, Wassim Swaileh, Shiyan Sun, and Ioannis Patras. ReWind: Understanding long videos with instructed learnable memory. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: Memory-augmented large multimodal model for long-term video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: OVBench and VideoChat-Online. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, 2020.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot-V: Memory-constrained KV cache compression for streaming video understanding. In *Advances in Neural Information Processing Systems*, 2025.

Jingdi Lei, Di Zhang, Junxian Li, Weida Wang, Kaixuan Fan, Xiang Liu, Qihan Liu, Xiaoteng Ma, Baian Chen, and Soujanya Poria. δ-mem: Efficient online memory for large language models. *arXiv preprint arXiv:2605.12357*, 2026.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. VideoChat-Flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, 2026.

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 2024b.

Bowen Liu, Shuning Wang, Xinpeng Ding, Zhiheng Wu, Bodong Du, and Xiaomeng Li. Beyond frame selection: Generative latent evidence aggregation for long-video understanding. *arXiv preprint arXiv:2607.28516*, 2026.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems*, 2023.

Zhenyu Ning, Guangda Liu, Qihao Jin, Chengwei Li, Wenchao Ding, Minyi Guo, and Jieru Zhao. LiveVLM: Efficient online video understanding via streaming-oriented KV cache and retrieval. *arXiv preprint arXiv:2505.15269*, 2025.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception Test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, 2023.

Minghao Qin, Xiangrui Liu, Zhengyang Liang, Yan Shu, Huaying Yuan, Junjie Zhou, Shitao Xiao, Bo Zhao, and Zheng Liu. Video-XL-2: Towards very long-video understanding through task-aware KV sparsification. *arXiv preprint arXiv:2506.19225*, 2025.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, 2021.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4, 1992.

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

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

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. MovieChat: From dense token to sparse memory for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. LVBench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. VideoLLaMB: Long streaming video understanding with recurrent memory bridges. In *IEEE/CVF International Conference on Computer Vision*, 2025.

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

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Haomiao Xiong, Zongxin Yang, Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Jiawen Zhu, and Huchuan Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. In *International Conference on Learning Representations*, 2025.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. SlowFast-LLaVA: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. StreamMem: Query-agnostic KV cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. In *AAAI Conference on Artificial Intelligence*, 2019.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *IEEE/CVF International Conference on Computer Vision*, 2025.

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

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018.

APPENDIX CONTENTS

A Reproducibility and Implementation Details p. 14
B Complete Experimental Results and Analyses p. 17
C Mathematical Notation Reference p. 21
D Limitations and Future Work p. 22
E LLM Usage p. 22

A REPRODUCIBILITY AND IMPLEMENTATION DETAILS

A.1 TRAINING CONFIGURATION AND HYPERPARAMETERS

All PREM variants utilize the bounded visual buffer formulation and spatially mean-pooled temporal feature projections (§3.1). The vision encoder and language model backbone remain frozen throughout training. We optimize solely PREM modules for one epoch using AdamW (Loshchilov & Hutter, 2019) (β_1=0.9, β_2=0.95, weight decay 0.01) with gradient norm clipping at 1.0. The peak learning rate is set to 2×10^{-4} for Qwen2.5-VL-3B and 10^{-4} for Qwen3-VL-8B, scheduled with a linear warmup for the first 3% of steps followed by cosine annealing to zero in bfloat16 mixed precision. Each training step processes an individual video-QA pair (per-GPU batch size of 1 without gradient accumulation). Distributed Data Parallelism (DDP) scales training across 8 NVIDIA A100 (80GB) GPUs, completing in ≈3.5 hours for Qwen2.5-VL-3B and ≈6.5 hours for Qwen3-VL-8B. The recurrent state (S, c) is initialized to zero and rebuilt independently per video instance. Videos are sampled uniformly at 1 FPS with frame resolutions dynamically constrained to at most 200,704 pixels (≈448 × 448). The training write cap is fixed to T=64, while the decoder prompt draws a visual buffer of B frames from the identical temporal sequence.

Trainable parameter counts are 9.07M for Qwen2.5-VL-3B and 27.48M for Qwen3-VL-8B, corresponding to ≈0.24% and ≈0.30% of the base architectures. The fp32 recurrent state occupies $4K(d_v d_k + 1)$ bytes: 262,160 bytes (≈256 KiB) for the standard K=4 configuration, and 65,540 bytes (≈64 KiB) for the K=1 controls in Table 2.

A.2 TRAINING DATA AND DECONTAMINATION PROTOCOL

The training corpus is a curated compact subset of LLaVA-Video-178K (Zhang et al., 2024), sourced from NextQA (Xiao et al., 2021), ActivityNet (Caba Heilbron et al., 2015; Yu et al., 2019), Charades (Sigurdsson et al., 2016), YouCook2 (Zhou et al., 2018), the Perception Test (Pătrăucean et al., 2023), and public YouTube video clips. Training uses 6,430 unique videos and 17,849 question-answer pairs from LLaVA-Video-178K (≈3.6% of videos and ≈1.4% of QA pairs). The full repository comprises 178K videos and 1.3M instruction pairs.

Decontamination. We filter training IDs that match the six evaluation benchmarks: LongVideoBench, MLVU, Video-MME, EgoSchema, MVBench, and LVBench. Decontamination filters out any training sample matching canonical YouTube video IDs, Ego4D universally unique identifiers (UUIDs), native benchmark keys, and Charades clip IDs reused by MVBench. Filename collisions across verified independent sources are treated as distinct videos. Table 3 details the resulting source composition, duration distribution, and sample counts.

A.3 BASELINE IMPLEMENTATION AND REPRODUCTION PROTOCOL

To ensure rigorous system comparisons in Table 1, we reproduce or benchmark external long-video architectures using their official open-source repositories under their recommended inference environments and official checkpoints:

- **LLaMA-VID** (Li et al., 2024b): Evaluated using the official Vicuna-7B checkpoint with 1 FPS uniform sampling and dual-token frame representation (generating 2 tokens per frame).
- **VideoLLaMA2** (Cheng et al., 2024): Evaluated using the official Qwen2-7B base checkpoint at B=16 uniformly sampled frames.
- **LLaVA-Video** (Zhang et al., 2024): Evaluated using the official 7B SFT checkpoint with B=64 frames under official prompt templates.
- **LongVILA** (Chen et al., 2024): Evaluated using the official long-context adapted Qwen2-7B checkpoint under official budget configs (B=32 or B=256 according to benchmark specifications).
- **LongVU** (Shen et al., 2025): Evaluated using the official Qwen2-7B SFT checkpoint with cross-frame spatio-temporal reduction at 0.5 FPS (and 2 FPS on short-video MVBench).
- **StreamChat** (Xiong et al., 2025): Evaluated training-free atop LongVA-7B, applying dynamic token pruning to retain 20% of source visual frames.

Table 3: **Training split composition after sampling and video-level decontamination.** Filtered from LLaVA-Video-178K (Zhang et al., 2024). Source and duration blocks summarize the retained split (6,430 unique videos and 17,849 QA pairs) with zero evaluation benchmark overlap.

Split Category	Primary Task / Domain	Videos	Share (%)	QA Pairs	Share (%)	QA / Vid
(a) Source Composition						
NextQA	Causal & temporal reasoning	2,886	44.9	5,785	32.4	2.0
ActivityNet	Long human action understanding	1,325	20.6	3,207	18.0	2.4
YouTube	Open-domain video reasoning	804	12.5	3,361	18.8	4.2
Charades	Indoor daily activity interaction	543	8.4	2,371	13.3	4.4
Perception Test	Fine-grained spatio-temporal perception	557	8.7	1,787	10.0	3.2
YouCook2	Procedural recipe & step grounding	315	4.9	1,338	7.5	4.2
(b) Video Duration Distribution						
0–30 s	Short clips (rapid actions)	2,135	33.2	6,013	33.7	2.8
30–60 s	Medium clips (event transitions)	2,267	35.3	6,468	36.2	2.9
1–2 min	Extended sequences (multi-stage)	1,382	21.5	3,379	18.9	2.4
2–3 min	Long horizon (narrative/procedural)	646	10.0	1,989	11.1	3.1
Total Retained	Multi-domain multimodal instruction	**6,430**	**100.0**	**17,849**	**100.0**	**2.8**

- **ReKV** (Di et al., 2025): Evaluated training-free on LLaVA-OneVision-7B at 0.5 FPS, using offline KV-cache construction with dynamic top-64 KV-chunk retrieval.
- **Flash-VStream** (Zhang et al., 2025): Evaluated using the official Qwen2-VL-7B LoRA checkpoint with a multi-resolution streaming memory buffer over $B{=}240$ ingested frames.
- **MovieChat** (Song et al., 2024): Evaluated training-free on Vicuna-7B with sliding-window temporal consolidation (128 video fragments $\times$ 8 frames).
- **Generalist Base VLMs**: Qwen2.5-VL-3B, and Qwen3-VL-8B are evaluated zero-shot via official implementations under greedy one-token constrained decoding.

A.4 Controlled Same-Backbone Mechanisms

Table 2 uses Qwen2.5-VL-3B with the registered $T{=}240$, $B{=}16$ streaming protocol, bfloat16 eager attention, and one-token constrained decoding. The backbone, resolution cap, frame sampling, evaluation manifests, and answer interface are fixed across rows. Base and LoRA consume the same deterministic B-frame reservoir; Token-Readout and PREM additionally build a recurrent state from the sampled stream. Token-Readout and PREM both use $K{=}1$, matching memory capacity in this controlled comparison. Adapted rows share this 6,430-video, 17,849-QA split and seed 13.

- **LoRA:** Freezes the vision encoder and language backbone except for self-attention projections q_{proj}, k_{proj}, v_{proj}, o_{proj} across all 36 transformer layers (rank $r{=}20$, $\alpha{=}40$, dropout 0, no bias adaptation; 9,216,000 trainable parameters). Trained for one epoch with AdamW at learning rate 2×10^{-4}, 3% warmup, global batch size 8. LoRA adds no prompt tokens and has no recurrent writer.
- **Token-Readout:** Reuses the exact same recurrent writer and trainable capacity as PREM ($K{=}1, d_v{=}d_k{=}128$, 4 layer groups, writer cap 128 tokens, $\lambda_{\text{pred}}{=}0.2$, 4 prediction targets; 9,070,982 trainable parameters), trained with the same optimizer schedule and cross-entropy plus prediction MSE. Instead of prefix K/V steering, the slot readout projects into $2K(1+4){=}10$ continuous tokens via global and group-specific heads, RMS-normalized to mean prompt norm and inserted at the prompt/answer boundary (labels masked, RoPE positions shifted).
- **InfiniPot-V:** Evaluated using the official InfiniPot-V algorithm adapted to Qwen2.5-VL-3B eager attention without fine-tuning, dynamically maintaining top-k informative KV caches across ingested chunks up to the target memory budget while evicting redundant spatiotemporal KV entries.

Cross-Family Training Details for LLaVA-Video-7B. For the LLaVA-Video-7B evaluation (Table 6), PREM modules ($K{=}4, d_v{=}d_k{=}128$, grouping $G{=}1$) are trained for one epoch on the identical decontaminated training split using AdamW with peak learning rate 1×10^{-4} (3% linear warmup, cosine decay), global batch size 8, and bfloat16 precision on 8$\times$A100 GPUs ($\approx$5.5 hours). Trainable parameters total 18.34M ($\approx$0.26% of the 7B backbone). The vision backbone and language layers remain completely frozen.

A.5 Decoder Token Order and Attention Interface

The decoder keeps the frozen backbone chat template. For Qwen2.5-VL and Qwen3-VL, the user turn places the video sequence immediately prior to the question and options; the processor expands video placeholders into the B-frame visual token sequence. LLaVA-Video-7B employs an equivalent image-token expansion. The resulting sequence structure is:

Instruction / System $\longrightarrow$ **B-Frame Visual Tokens** $\longrightarrow$ **Question & Options** $\longrightarrow$ **Answer Token**

Text Tokens $\longrightarrow$ Visual Patch Tokens $\longrightarrow$ Text Prefix Tokens $\longrightarrow$ Constrained Decoding

Visual token spans are identified via backbone video/image placeholder indices. The steering mask $\mathcal{M}_{\text{prefix}}$ is the complement of this visual span, intersected with the valid attention mask and restricted to prompt prefix positions preceding the answer generation token. Writer inputs x_t are spatially mean-pooled visual features extracted from the frozen vision encoder, outside the decoder sequence. PREM executes natively within the backbone causal self-attention kernel. The implementation adds the computed prefix modulations $\Delta K_i^{(\ell)}$ and $\Delta V_i^{(\ell)}$ to the outputs of the frozen K/V projection hooks:

$$\widetilde{K}_i^{(\ell)} = W_K^{(\ell)} h_i^{(\ell)} + \Delta K_i^{(\ell)}, \qquad \widetilde{V}_i^{(\ell)} = W_V^{(\ell)} h_i^{(\ell)} + \Delta V_i^{(\ell)},$$

For Qwen3-VL, Q/K RMSNorm follows the projection hooks; RoPE then rotates Q/K before attention and cache storage. Qwen2.5-VL omits Q/K RMSNorm. Consequently, Qwen3-VL's effective key perturbation is nonlinear, while its value perturbation remains additive. Visual tokens, padding tokens, and answer positions are unsteered (d_i=0). During autoregressive generation, modulation operates exclusively during prefill; the decoding cache directly serves the steered prefix representations without recomputation.

A.6 State Initialization, Constants, and Training Batch

Table 4 documents the initialization values and fixed hyperparameters for Equations 4–6. For every video, (S, c) starts at zero. Streaming end-of-stream evaluation carries (S, c) across consecutive two-frame chunks and answers after video completion. Over n steps without writes, each slot retains a factor $\sigma(f_m)^n$ of its state, while confidence decays by γ_c^n. At initialization ($f_m = 4$), state retention after 240 steps is approximately 1.3%; training learns f_m. The extra $b_{t,m}$ factor discounts unstable writes in the confidence update; the $1/K$ factor in Eq. 7 damps the routed steering amplitude as slot count increases. The forget logit f_m is learned via backpropagation, whereas the write step size η_w, confidence decay γ_c, and read-router temperature τ are held constant. The write router operates with an untempered softmax, while temperature τ=1.0 governs the read router in Equation 6. The auxiliary load-balancing weight is λ_{bal}=0.10 by default (swept in Table 10). Main checkpoints use K=4 and d_v=d_k=128; the slot and layer-group sweeps vary K and G, respectively.

A.7 Feature Prediction Head

For prediction, we select a contiguous $\min(T_{\text{pred}}, T_w - 1)$-step span with the highest detached mean salience, block its writes, and run a second writer pass to obtain a held-out state S^-. Let T_w be the pooled writer sequence length, $\tilde{x}_t = \text{LN}(x_t) \in \mathbb{R}^{d_{\text{model}}}$ denote the video-only writer input, $q \in \mathbb{R}^{d_{\text{model}}}$ the mean non-visual prefix embedding, and $\bar{t} = (t-1)/(T_w - 1) \in [0, 1]$ the normalized temporal position. The prediction read reuses W_r, A_m, W_{qv}, and R from Eqs. 5–6, with $W_r\,\text{LN}(q)$ as its query. The prediction branch reads the held-out state at every writer step and computes

$$\begin{aligned} r_t^- &= \text{Read}(S^-, q\mathbf{1}_{T_w})_t \in \mathbb{R}^{d_v}, \\ h_t &= r_t^- + P_q\,\text{LN}(q) + P_p\bar{t} + b_p, \\ \hat{y}_t &= F_{\text{pred}}(h_t), \end{aligned} \tag{10}$$

Here $P_q \in \mathbb{R}^{d_v \times d_{\text{model}}}$ is bias-free, $P_p \in \mathbb{R}^{d_v \times 1}$ has bias $b_p \in \mathbb{R}^{d_v}$, and F_{pred} is $\text{LN}_{d_v} \to \text{Linear}(d_v, 2d_v) \to \text{GELU} \to \text{Linear}(2d_v, d_v)$. The target is the stop-gradient, normalized value projection $y_t = \text{sg}(\text{norm}(W_v\tilde{x}_t))$, and the masked objective is

$$\mathcal{L}_{\text{pred}} = \frac{1}{\sum_t m_t} \sum_{t=1}^{T_w} m_t \|\text{norm}(\hat{y}_t) - y_t\|_2^2, \tag{11}$$

where m_t is the selected span mask. The loss is evaluated per video and averaged across the batch; sequences with $T_w \leq 1$ contribute zero. Thus the head has input and output shape $[N, T_w, d_v]$ for batch size N; its only temporal conditioning is the scalar normalized position $\bar{t}$, while query conditioning is broadcast across the T_w writer steps. The branch contains $d_v d_{\text{model}} + 4d_v^2 + 7d_v$

Table 4: **Initialization and operational constants for PREM.** The standard configuration uses $K{=}4$ and $d_v{=}d_k{=}128$. Recurrent states reset independently per video instance; parameters are optimized via AdamW or maintained as fixed constants.
Default linear weights and biases use $\mathcal{U}(-1/\sqrt{d_{\text{in}}}, 1/\sqrt{d_{\text{in}}})$. LayerNorm uses unit scale, zero bias, and $\epsilon_{\text{LN}} = 10^{-5}$.

Quantity / Symbol	Initialization / Value	Operational Status
Recurrent Memory State Variables		
Matrix state $S \in \mathbb{R}^{K \times d_v \times d_k}$	Zero tensor ($\mathbf{0}$)	Update (Eq. 4)
Confidence vector $c \in [0,1]^K$	Zero vector ($\mathbf{0}$)	Update (Eq. 4)
Trainable Model Parameters (AdamW Optimization)		
Forget logit f_m	4.0 ($\sigma(f_m) \approx 0.982$)	Trainable
Slot address transform A_m	$I + 0.02\,\mathcal{N}(0,1)$	Trainable
Visual projections W_k, W_v	$d_k \times d_{\text{model}}$ and $d_v \times d_{\text{model}}$; rectangular near-identity $+0.01\,\mathcal{N}(0,1)$	Trainable
Write routing r_w	$\texttt{Linear}(d_k, K)$, default initialization	Trainable
Update stability W_b	$\texttt{Linear}(d_k, K)$, default initialization	Trainable
Visual salience ϕ	$\texttt{Linear}(d_{\text{model}}, \lfloor d_{\text{model}}/4 \rfloor)$–GELU–$\texttt{Linear}(\lfloor d_{\text{model}}/4 \rfloor, 1)$, default initialization	Trainable
Read conditioning W_c	$d_{\text{model}} \times d_{\text{model}}$, bias-free; default initialization	Trainable
Read projection W_r	$d_k \times d_{\text{model}}$, copied from the initialized W_k at construction	Trainable
Query–value alignment W_{qv}	$d_v \times d_k$, bias-free; default initialization	Trainable
Read router R	$\texttt{Linear}(3d_v, d_v)$–GELU–$\texttt{Linear}(d_v, 1)$; $3d_v^2 + 2d_v + 1$ parameters (49,409 at $d_v{=}128$), default initialization	Trainable
Steering projection heads $H_{g,m}^{k,v}$	$10^{-4}\mathcal{N}(0,1)$	Trainable
Base steering scale $\hat{\alpha}$	$\log \alpha_{\text{train}}$	Trainable
Prediction branch $(P_q, P_p, F_{\text{pred}})$	$d_v d_{\text{model}} + 4d_v^2 + 7d_v$; P_q near-identity $+0.01\,\mathcal{N}(0,1)$, P_p/F_{pred} default Linear/LayerNorm init; 328,576 for Qwen2.5-VL-3B	Learned only when $\lambda_{\text{pred}} > 0$
Fixed Hyperparameters (Frozen Architectural Constants)		
Base write step size η_w	1.0	Frozen constant
Confidence momentum γ_c	0.95	Frozen constant
Read-router temperature τ	1.0	Frozen constant

trainable parameters (328,576 for Qwen2.5-VL-3B with $d_{\text{model}}{=}2048, d_v{=}128$). It is used only during training and is disabled at test time.

A.8 Evaluation Protocols and Benchmark Specifications

Evaluation is conducted at 1 FPS with a maximum resolution cap of 200,704 pixels per frame. For videos up to 240 seconds, frames are sampled at 1 FPS; for longer videos, the stream is uniformly sampled across the full temporal horizon up to the maximum ingestion cap of $T{=}240$ frames. Both the writer and the decoder share the same frame sample under identical visual resolution caps. Testing evaluates decoder visual budgets $B \in \{16, 32, 64, 90\}$; same-backbone comparisons match B identically. In offline mode, the visual prompt buffer $\mathcal{V}_B$ samples B frames uniformly across the stream, and the writer processes the sequence in a single pass. In streaming end-of-stream mode, frames arrive chronologically in sequential two-frame chunks carrying state (S, c) forward while a deterministic uniform B-frame reservoir is maintained; the reservoir contents form the prompt buffer when the stream ends. All benchmarks evaluate via one-token constrained decoding over option letters $\{A, B, C, D, \dots\}$ using official evaluation scripts across 12,261 total questions: LongVideoBench (1,337), MLVU (2,175), Video-MME (2,700), EgoSchema (500), MVBench (4,000), and LVBench (1,549).

B Complete Experimental Results and Analyses

This section provides the exhaustive tabular results and fine-grained breakdowns supporting the analyses in Section 4. Tables 5 and 6 present the complete per-benchmark results across decoder budgets $B \in \{16, 32, 64, 90\}$ for Qwen and LLaVA-Video backbones (supporting Figure 4). Table 8

documents the individual benchmark accuracies across steering targets (supporting Figure 6a). Table 9 reports per-benchmark accuracies across writer training horizons. Table 7 reports streaming end-of-stream latency and memory. Furthermore, Figure 11 and Table 10 document exhaustive sensitivity sweeps across 36 parameter configurations and a clean Base reference.

B.1 Diagnostic Configurations

The duration analysis in Figure 5 uses the T_{mem}=16 checkpoint from Table 10: K=4, G=1, training frame cap 64, α_{train}=α_{run}=1, λ_{bal}=0.05, λ_{pred}=0.1, and T_{pred}=8. Its aggregate accuracies are 54.56% on Video-MME and 38.09% on LVBench. The diagnostic compares the uniform-frame Base with a reservoir-buffer PReM run at B=16 and ingestion cap 240. The component ablations in Table 8 share the Full PReM setup except for the stated component change. Table 10 varies its stated factor within each sweep.
Table 7 reports runtime under T=240 evaluation ingestion from separate profiling runs; accuracies follow Table 5. Latency columns are rounded independently.

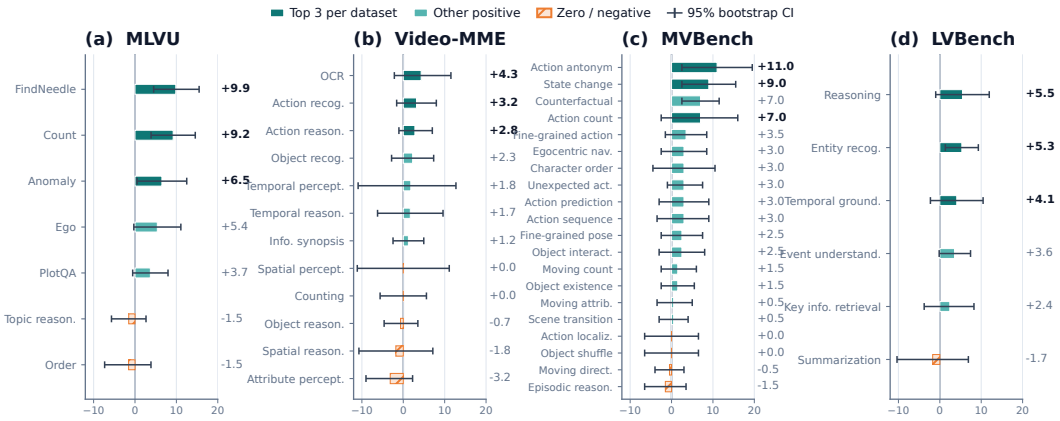

Figure 10: Exhaustive native task-category accuracy differences (PReM − Base, %) on Qwen2.5-VL-3B at offline B=16. Error bars denote 95% paired bootstrap confidence intervals across question resamples. Benchmarks lacking native task categories (LongVideoBench, EgoSchema) are omitted.

Table 5: **Matched decoder budgets on Qwen backbones.** Accuracy (%). Each cell shows Base → PReM performance across B. Bold highlights the superior score between paired settings; Δ denotes absolute Avg gain. B=240 is a high-budget Base reference.

Budget	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg	Δ
(a) Qwen2.5-VL-3B (Offline Ingestion)								
B=16	49.06 → **53.48**	54.90 → **59.82**	53.11 → **54.19**	54.40 → **54.60**	56.50 → **60.13**	34.34 → **38.48**	50.39 → **53.45**	↑3.06
B=32	50.86 → **53.33**	58.53 → **62.48**	57.26 → **57.44**	56.80 → **58.60**	57.77 → **61.50**	37.70 → **40.03**	53.15 → **55.56**	↑2.41
B=64	51.68 → **55.65**	61.47 → **64.23**	59.56 → **59.81**	60.60 → **61.40**	57.88 → **61.30**	38.99 → **41.06**	55.03 → **57.24**	↑2.21
B=90	52.13 → **55.27**	63.45 → **65.47**	61.26 → **61.30**	61.20 → 61.20	57.92 → **61.35**	40.41 → **43.12**	56.06 → **57.95**	↑1.89
Ref (B=240)	53.78	65.61	62.15	62.20	60.50	43.12	57.89	–
(b) Qwen2.5-VL-3B (Streaming End-of-Stream)								
B=16	52.73 → **53.55**	55.68 → **59.68**	53.89 → **54.11**	54.40 → **54.60**	59.98 → **60.33**	35.38 → **38.48**	52.01 → **53.46**	↑1.45
B=32	53.40 → **54.53**	60.00 → **62.34**	56.26 → **57.70**	58.40 → **58.60**	61.38 → **61.50**	37.90 → **40.09**	54.56 → **55.79**	↑1.23
B=64	54.90 → **56.02**	61.66 → **64.18**	58.93 → **59.70**	60.60 → **60.80**	61.08 → **61.33**	41.12 → **41.45**	56.38 → **57.25**	↑0.87
B=90	54.75 → **56.54**	63.59 → **65.52**	59.63 → **61.30**	**61.40** → 61.20	61.20 → **61.40**	41.77 → **42.93**	57.06 → **58.15**	↑1.09
(c) Qwen3-VL-8B (Offline Ingestion)								
B=16	56.84 → **57.52**	60.55 → **62.30**	60.00 → **60.41**	69.00 → **70.00**	64.60 → **66.15**	38.00 → **38.28**	58.16 → **59.11**	↑0.95
B=32	58.94 → **59.69**	65.15 → **65.29**	63.33 → **64.04**	70.00 → **73.00**	65.38 → **67.33**	39.83 → **40.67**	60.44 → **61.67**	↑1.23
B=64	60.36 → **61.85**	68.18 → **70.30**	66.11 → **66.96**	70.60 → **74.60**	65.83 → **67.03**	41.83 → **44.16**	62.15 → **64.15**	↑2.00
B=90	60.88 → **63.05**	70.94 → **71.22**	67.56 → **67.87**	72.60 → **74.20**	65.90 → **67.10**	45.32 → **45.76**	63.87 → **64.87**	↑1.00
Ref (B=240)	64.25	75.59	70.19	73.00	65.95	47.39	66.06	–
(d) Qwen3-VL-8B (Streaming End-of-Stream)								
B=16	56.92 → **57.74**	59.17 → **62.53**	58.41 → **60.52**	69.00 → **69.40**	64.55 → **66.13**	36.99 → **38.22**	57.51 → **59.09**	↑1.58
B=32	58.56 → **60.58**	62.90 → **65.29**	62.52 → **64.00**	71.40 → **72.40**	65.83 → **67.25**	40.48 → **40.67**	60.28 → **61.70**	↑1.42
B=64	**62.00** → 61.78	67.26 → **70.34**	65.67 → **66.96**	72.20 → **74.60**	66.03 → **67.00**	42.87 → **43.71**	62.67 → **64.07**	↑1.40
B=90	61.71 → **62.15**	69.75 → **71.22**	67.22 → **67.87**	72.40 → **74.20**	66.05 → **67.13**	42.61 → **45.76**	63.29 → **64.72**	↑1.43

Table 6: **Cross-family evaluation on LLaVA-Video-7B.** Accuracy (%). Each cell shows Base → PREM performance across decoder budgets B. Bold highlights the superior score between paired settings; Δ denotes absolute Avg gain over Base. Evaluated through budget B=90.

Budget	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg	Δ
(a) LLaVA-Video-7B (Offline Ingestion)								
B=16	56.32 → **57.52**	61.29 → **62.76**	59.96 → **60.07**	54.20 → **56.00**	58.70 → **59.03**	38.35 → **38.80**	54.80 → **55.70**	↑**0.90**
B=32	58.12 → **58.64**	66.57 → **66.76**	62.15 → **62.37**	58.00 → **58.20**	59.90 → **60.48**	40.87 → **42.29**	57.60 → **58.12**	↑**0.52**
B=64	59.39 → **59.99**	70.02 → **70.39**	63.33 → **63.48**	**58.20** → 58.00	59.60 → **60.15**	42.41 → **43.58**	58.82 → **59.27**	↑**0.45**
B=90	59.69 → **60.66**	70.39 → **70.57**	**63.74** → 63.52	58.40 → **59.00**	60.12 → **60.18**	45.13 → **45.47**	59.58 → **59.90**	↑**0.32**
(b) LLaVA-Video-7B (Streaming End-of-Stream)								
B=16	55.87 → **56.47**	60.46 → **61.61**	59.41 → **60.07**	53.80 → **54.00**	57.43 → **58.83**	39.57 → **40.28**	54.42 → **55.21**	↑**0.79**
B=32	57.29 → **58.12**	65.75 → **66.11**	**61.00** → 60.96	**57.60** → 57.00	58.03 → **58.25**	40.48 → **40.87**	56.69 → **56.89**	↑**0.20**
B=64	57.29 → **58.79**	68.18 → **69.66**	62.59 → **63.00**	58.20 → **58.40**	58.03 → **58.18**	41.83 → **43.58**	57.69 → **58.60**	↑**0.91**
B=90	58.04 → **59.61**	68.55 → **70.57**	63.85 → **63.98**	59.00 → **59.60**	58.05 → **58.90**	42.16 → **44.74**	58.28 → **59.57**	↑**1.29**

Table 7: **Streaming end-of-stream efficiency on Qwen2.5-VL-3B.** Evaluated on NVIDIA H20 (1 FPS, batch size 1), supporting Figure 9c–d. Base and PREM share decoder budget B. Ingestion and answering are averaged over all QA pairs, including cache hits with negligible ingestion time; total latency reports mean / p95. Subscripts indicate peak memory overhead; Δ denotes absolute Avg gain.

Method	B	Backbone / Train. Params (M)	Peak Mem. (GiB)	Ingest / Ans. (s)	Mean / p95 (s)	Avg	Δ
Base	16	3754.62 / 0.00	7.43	5.64 / 0.30	5.94 / 29.78	52.01	–
PREM	16	3754.62 / 9.07	$7.46_{+0.03}$	7.81 / 0.38	8.19 / 38.11	**53.46**	↑**+1.45**
Base	32	3754.62 / 0.00	7.83	5.71 / 0.53	6.24 / 30.31	54.56	–
PREM	32	3754.62 / 9.07	$7.86_{+0.03}$	7.83 / 0.64	8.48 / 38.76	**55.79**	↑**+1.23**
Base	64	3754.62 / 0.00	8.64	5.81 / 0.96	6.77 / 31.44	56.38	–
PREM	64	3754.62 / 9.07	$8.67_{+0.03}$	7.88 / 1.15	9.02 / 40.02	**57.25**	↑**+0.87**
Base	90	3754.62 / 0.00	9.29	5.89 / 1.31	7.19 / 32.48	57.06	–
PREM	90	3754.62 / 9.07	$9.32_{+0.03}$	7.90 / 1.55	9.45 / 40.95	**58.15**	↑**+1.09**

Table 8: **Diagnostic component ablations on Qwen2.5-VL-3B** (offline B=16, %). Steering targets evaluate separately trained heads. All variants share the same experimental setup except for the ablated component. Δ denotes the score drop relative to Full PREM. Zero-buffer probes (B=0) isolate recurrent retention without visual prompts.

Variant	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg	Δ vs. Full
Base (B=16, no memory)	49.06	54.90	53.11	54.40	56.50	34.34	50.39	–
Full PREM (K/V modulation)	**53.48**	59.82	**54.19**	**54.60**	**60.13**	38.48	**53.45**	–
Modulation Interface								
Q/O modulation	52.66	59.49	53.81	47.00	59.98	36.02	51.49	↓1.96
Key-only (K) modulation	52.28	56.14	53.33	52.00	59.78	36.41	51.66	↓1.79
Value-only (V) modulation	51.91	59.86	53.59	49.80	59.12	37.25	51.92	↓1.53
Write Gating Terms								
Uniform write routing (π^w)	52.73	59.36	53.96	52.40	59.58	37.96	52.66	↓0.79
w/o Update stability (b)	52.06	57.70	53.48	53.00	59.60	37.70	52.26	↓1.19
w/o Visual salience (z)	51.91	58.53	54.00	50.80	59.73	37.83	52.13	↓1.32
w/o Confidence protection (a)	52.43	**60.23**	53.70	49.40	59.45	37.44	52.11	↓1.34
w/o Residual novelty (n)	51.98	59.17	53.30	49.60	59.38	**38.73**	52.03	↓1.42
Zero-Buffer Retention Probes (B=0)								
PREM (B=0, memory only)	43.40	47.73	40.65	27.80	40.88	29.50	38.33	–
Base (B=0, vision-free)	39.27	42.02	37.30	26.80	36.70	26.86	34.82	–

Table 9: **Writer training budget.** Per-benchmark accuracy (%) on Qwen2.5-VL-3B (B=16, evaluated at T=240), supporting Figure 9a–b. Checkpoints use write cap T (α_{run}=0.75); clean Base rows provide the protocol-matched reference. Shading and bold highlight the highest Avg per protocol.

Training Horizon	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg	Δ
Offline Ingestion								
Clean Base	49.06	54.90	53.11	54.40	56.50	34.34	50.39	–
Writer T=16	52.58	58.25	53.85	52.80	59.92	37.96	52.56	↑**+2.17**
Writer T=32	52.58	57.75	53.78	52.20	59.98	38.22	52.42	↑**+2.03**
Writer T=64	53.48	59.82	54.19	54.60	60.13	38.48	**53.45**	↑**+3.06**
Writer T=90	53.18	59.82	54.04	49.80	60.12	37.44	52.40	↑**+2.01**
Writer T=240	52.36	58.99	54.07	52.40	59.52	38.48	52.64	↑**+2.25**
Streaming End-of-Stream								
Clean Base	52.73	55.68	53.89	54.40	59.98	35.38	52.01	–
Writer T=16	51.91	58.11	53.85	52.60	60.15	37.96	52.43	↑**+0.42**
Writer T=32	52.36	58.02	53.52	52.40	60.33	38.09	52.45	↑**+0.44**
Writer T=64	53.55	59.68	54.11	54.60	60.33	38.48	**53.46**	↑**+1.45**
Writer T=90	53.25	59.68	53.85	50.40	60.23	37.38	52.46	↑**+0.45**
Writer T=240	51.38	59.17	54.00	51.80	59.70	37.96	52.34	↑**+0.33**

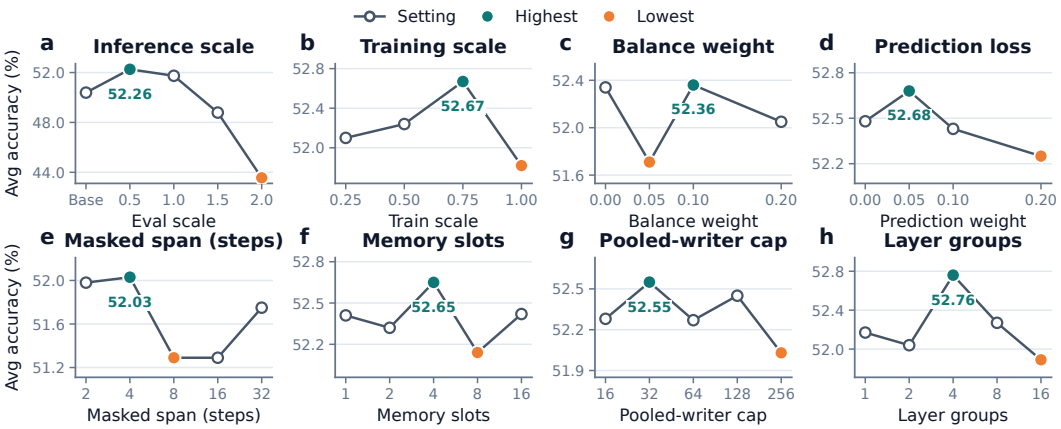

Figure 11: One-factor sensitivity sweeps on Qwen2.5-VL-3B (offline B=16, Avg). Markers denote checkpoints or inference configurations; teal and orange highlight extreme values within each sweep. Corresponding per-benchmark numeric breakdowns are tabulated in Table 10.

Table 10: **Hyperparameter sensitivity on Qwen2.5-VL-3B** (offline B=16, %; configurations in Appendix B.1). (a) Training sweeps: α_{train}, λ_{bal}, λ_{pred}, and T_{pred} (writer steps); read-router temperature τ=1 is fixed. (b) Inference scale α_{run} changes steering without retraining; Base is the clean frozen reference. (c) Capacity sweeps: K slots, training-time pooled-writer cap T_{mem} (no prompt positions), and G layer-head groups. Shading and bold mark the highest Avg within each sweep.

Parameter	Setting	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg
(a) Training								
Training steering scale α_{train}								
	0.25	52.06	56.87	53.30	53.20	60.00	37.19	52.10
	0.50	52.95	57.52	54.07	51.80	59.68	37.44	52.24
	0.75	52.95	59.54	54.04	51.40	59.65	38.41	**52.67**
	1.00	51.98	60.18	53.74	48.60	58.73	37.70	51.82
Load-balancing weight λ_{bal}								
	0.00	52.36	60.55	54.15	49.40	59.30	38.28	52.34
	0.05	50.93	59.86	54.07	48.60	59.40	37.38	51.71
	0.10	53.03	59.54	53.81	51.20	59.20	37.38	**52.36**
	0.20	51.68	59.26	53.85	50.80	59.40	37.31	52.05
Prediction loss weight λ_{pred}								
	0.00	53.33	58.21	54.00	51.80	59.55	38.02	52.48
	0.05	53.03	58.16	54.00	52.40	59.88	38.61	**52.68**
	0.10	52.88	57.20	53.89	53.00	59.95	37.64	52.43
	0.20	52.21	56.97	53.67	53.00	59.95	37.70	52.25
Masked prediction span T_{pred} (writer steps)								
	2	52.80	59.13	53.26	49.60	59.63	37.44	51.98
	4	52.95	59.86	53.93	48.80	59.30	37.31	**52.03**
	8	52.13	59.72	53.04	46.20	59.65	36.99	51.29
	16	51.68	60.00	52.89	47.20	59.23	36.73	51.29
	32	51.91	59.68	52.78	49.40	59.38	37.38	51.75

Parameter	Setting	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg
(b) Inference								
Inference steering scale α_{run}								
	Base	49.06	54.90	53.11	54.40	56.50	34.34	50.39
	0.5	52.73	59.22	53.78	50.00	59.88	37.96	**52.26**
	1.0	52.51	60.32	53.07	47.20	59.63	37.70	51.74
	1.5	49.21	56.87	50.89	39.80	58.73	37.25	48.79
	2.0	44.80	47.26	46.19	35.40	55.63	32.09	43.56
(c) Memory capacity								
Memory slot count K								
	1	51.76	58.90	54.11	52.60	59.33	37.77	52.41
	2	52.43	59.63	54.37	50.20	59.33	37.96	52.32
	4	52.80	58.67	54.26	52.20	59.78	38.22	**52.65**
	8	53.25	58.57	53.59	50.00	59.23	38.22	52.14
	16	52.13	57.89	53.48	54.20	59.78	37.06	52.42
Pooled-writer cap T_{mem}								
	16	52.73	59.31	54.56	49.20	59.78	38.09	52.28
	32	52.88	59.36	53.89	52.40	59.88	36.86	**52.55**
	64	53.03	59.68	54.19	49.80	59.85	37.06	52.27
	128	52.95	59.26	54.11	51.80	59.53	37.06	52.45
	256	52.06	59.40	53.74	50.40	59.03	37.57	52.03
Layer-head groups G								
	1	53.03	58.99	54.19	49.20	59.50	38.09	52.17
	2	52.95	58.48	53.67	50.00	59.33	37.83	52.04
	4	53.78	59.22	53.96	52.60	59.68	37.31	**52.76**
	8	52.73	58.99	53.11	51.80	59.95	37.06	52.27
	16	52.13	58.62	53.00	51.40	59.90	36.28	51.89

B.2 Multi-seed Training Stability

To verify optimization robustness, Table 11 reports fine-grained per-benchmark accuracy alongside aggregate statistics across three independent training runs (seeds 13, 17, 23) at offline B=16. Seed 13 reuses the Table 5 main-run checkpoint. On Qwen2.5-VL-3B, mean accuracy is 53.38% with sample standard deviation (SD) 0.06%, a $+2.99 \pm 0.06\%$ gain over Base (50.39%); on Qwen3-VL-8B, the mean is 59.11% (SD=0.03%), a $+0.95 \pm 0.03\%$ gain over Base (58.16%). Every seed exceeds its paired Base Avg; Table 11 reports the per-benchmark variation.

Table 11: **Multi-seed evaluation across benchmarks** (offline B=16, %). Accuracy and standard deviations across three independent runs (seeds 13, 17, 23) alongside deterministic Base.

Method	Seed	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg
(a) Qwen2.5-VL-3B								
Base	–	49.06	54.90	53.11	54.40	56.50	34.34	50.39
PREM	13	53.48	59.82	54.19	54.60	60.13	38.48	53.45
PREM	17	53.33	59.54	54.15	54.20	60.15	38.67	53.34
PREM	23	53.10	60.64	53.81	54.00	59.90	38.67	53.35
Mean		53.30	60.00	54.05	54.27	60.06	38.61	**53.38**
SD (±)		0.19	0.57	0.21	0.31	0.14	0.11	0.06

Method	Seed	LVB	MLVU	VMME	EgoS	MVB	LVBN	Avg
(b) Qwen3-VL-8B								
Base	–	56.84	60.55	60.00	69.00	64.60	38.00	58.16
PREM	13	57.52	62.30	60.41	70.00	66.15	38.28	59.11
PREM	17	57.67	62.34	60.44	69.80	66.25	38.35	59.14
PREM	23	57.52	62.21	60.33	69.80	66.12	38.54	59.09
Mean		57.57	62.28	60.39	69.87	66.17	38.39	**59.11**
SD (±)		0.09	0.07	0.06	0.12	0.07	0.13	0.03

C Mathematical Notation Reference

Table 12 summarizes the mathematical symbols, dimensions, and operational definitions.

Table 12: **Mathematical notation reference for Prefix-Steered Recurrent Memory.** Summarizes the core symbols, tensor dimensions, and operational definitions of PREM across memory updates, prefix steering, and optimization objectives.

Symbol	Meaning and Operational Scope
Visual Input and Recurrent Memory State	
$E(v_t)_p,\ x_t$	Encoder patch token at position p and spatially pooled visual frame feature at time t
$S \in \mathbb{R}^{K \times d_v \times d_k}$	Recurrent associative memory state tensor across K memory slots

Table 12 (continued)

Symbol	Meaning and Operational Scope
$c \in [0,1]^K$	Per-slot write-confidence vector tracking temporal update history
$B,\ T$	Decoder prompt visual frame budget and total writer ingestion cap (train T=64; eval T=240)
Memory Write Dynamics and Gating System	
W_k, W_v, A_m	Learnable key/value projection matrices and slot-specific addressing transforms
$\kappa_{t,m}$	Normalized write addressing query for memory slot m at time t
$e_{t,m}$	Associative reconstruction residual $u_t - S_m \kappa_{t,m}$
$b_{t,m},\ n_{t,m}$	Update-stability gate and residual-novelty gate activations (Eq. 3)
$z_t,\ a_{t,m}$	Visual-salience gate and confidence-protection gate activations (Eq. 3)
$g_{t,m}$	Composite write coefficient: $\pi^w_{t,m} b_{t,m} n_{t,m} z_t a_{t,m}$
$\sigma(f_m),\ \eta_w,\ \gamma_c$	Learnable slot retention probability, base write step size, and confidence decay rate
π^w	Softmax write-routing distribution across associative slots
Prefix Steering and Read Interface	
$\xi^{(\ell)}_{i,m}$	Layer-wise normalized read query for slot m at prompt token index i
$\rho^r,\ \tau$	Normalized read relevance scores and read-router temperature parameter
$H^k_{g,m},\ H^v_{g,m}$	Group- and slot-specific key and value prefix steering projection heads
$d^{k,(\ell)}_i,\ d^{v,(\ell)}_i$	Additive prefix modulation vectors injected into backbone Key and Value paths
$W^{(\ell)}_K,\ W^{(\ell)}_V$	Frozen language model self-attention Key and Value projection weights
$\hat{\alpha},\ \alpha_{\text{run}},\ \alpha_{\text{eff}}$	Learnable scale logit, inference-time steering multiplier, and effective steering scale
$G,\ T_{\text{mem}}$	Attention layer group count and training-time pooled-writer cap
Training Objectives and Hyperparameters	
$\mathcal{L}_{\text{CE}}$	Primary cross-entropy loss over target response tokens
$\mathcal{L}_{\text{bal}},\ \lambda_{\text{bal}}$	Router load-balancing loss (uniformity penalty) and auxiliary loss weight
$\mathcal{L}_{\text{pred}},\ \lambda_{\text{pred}}$	Salient feature reconstruction loss and corresponding loss weight
T_{pred}	Masked prediction span measured in pooled writer steps

D Limitations and Future Work

Three design choices define the scope of this work:

- **Fixed Slot Granularity:** The recurrent memory uses a fixed number K of associative slots with fixed hidden dimensionality. Dynamic slot allocation or hierarchical multi-scale abstractions would capture complex event boundaries in continuous streams.
- **Frame Sampling Rate:** The architecture samples at 1 FPS, uniformly subsampling longer videos to T=240 frames. Spatial mean-pooling retains frame summaries while discarding intra-frame layout from the recurrent state. Dense action streams at higher native frame rates call for adaptive write filtering.
- **Evaluation Paradigms:** Evaluation targets offline and streaming end-of-stream multiple-choice QA; future work will extend PREM to open-ended dialogue and lifelong streaming agent environments.

E LLM Usage

Large language models assisted with manuscript critique, methodology and experiment feedback, language editing, LaTeX grammar checking, and text formatting. The conceptual methodology, mathematical formulation, model implementation, and experimental evaluations were conducted independently by the authors. All AI-assisted text, empirical results, derivations, and factual claims were verified against the source code, training logs, and evaluation checkpoints. The authors assume full responsibility for the contents of this paper.