STATEVID: VIDEO REPRESENTATION AS AN EVOLVING VISUAL STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Video encoders usually represent a video as a sequence of spatial features produced from individual frames. This frame-based representation treats each observation as a separate feature set, leaving the integration of persistent and changing visual content to later processing. We view frames as discrete observations of an evolving visual process, and its evolving state as the object of video representation. We propose StateVid, which updates this state causally at each spatial location with learned retention and writing gates. The state itself carries the video representation, and ordered snapshots expose its evolution to downstream models. The resulting visual state keeps a fixed size while accumulating evidence over time. StateVid learns these transitions from short video-text pairs through supervision of historical retention and adaptation to new content. StateVid demonstrates strong performance on video tasks compared to models trained at larger scales, while maintaining image capabilities. At the same time, it achieves better token efficiency and performance on VideoLLM. We additionally find that this representation method has stable extrapolation performance and exhibits selective attention to video content.

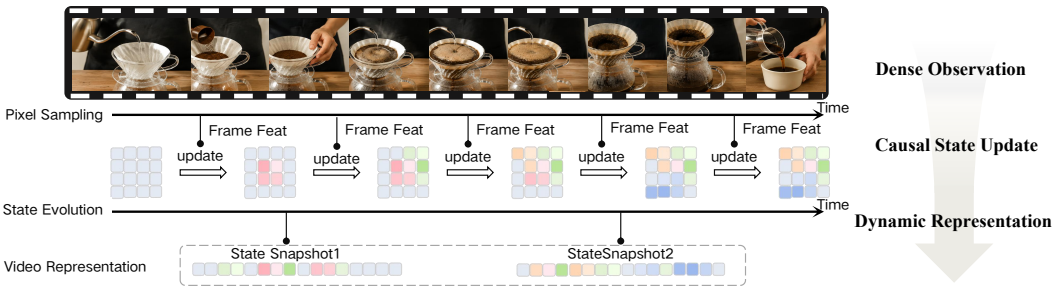

Figure 1: Overview of StateVid. Each sampled frame causally updates a persistent visual state through learned retention and writing gates. The evolving state accumulates video history, and snapshots selected along its trajectory form the representation passed to downstream models.

1 INTRODUCTION

Recent advances in video pretraining and multimodal learning have substantially improved video understanding (Wang et al., 2022; Li et al., 2023; Wang et al., 2024). Models can now recognize actions, understand events, and reason over visual context (Maaz et al., 2024; Zhang et al., 2025; 2023). As video understanding expands from short clips to more complex content, how visual evidence is organized over time becomes increasingly important. Many approaches build on spatial features extracted from individual frames. Video encoders introduce temporal interactions among these features (Fang et al., 2021; Ma et al., 2022; Lin et al., 2022; Bertasius et al., 2021; Arnab et al., 2021), while many VideoLLMs rely on the language model to relate frame representations (Zhang et al., 2024b; Li et al., 2024a; Zhang et al., 2025). Although temporal modeling occurs at different stages, the downstream representation is often still organized as spatial tokens indexed by observation time.

A frame-indexed interface requires the model to infer temporal persistence, visual change, and historical relevance from a growing collection of observations. Existing approaches address this chal-

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.

lenge by sampling observations, aggregating frame features (Luo et al., 2026; Li et al., 2026), or constructing memory after frame encoding (Song et al., 2024; He et al., 2024; Shen et al., 2024). Even when temporal interactions or memory are introduced, the downstream interface often remains organized around frame-indexed outputs. This motivates changing what the encoder represents. An effective video representation should directly expose what persists, what changes, and what remains relevant over time, so downstream models do not need to reconstruct these relationships from a growing sequence of frame-indexed features.

To meet these requirements, **we argue that video frames are discrete observations of an evolving visual process. The evolution of this underlying visual process can therefore serve as a compact representation of the video.** This perspective leads us to represent a video as an evolving visual process. As illustrated in Fig. 1, observations enter the state sequentially, the state evolves along the video, and ordered snapshots along this trajectory provide representations for downstream tasks. Each new observation changes the content of the state while preserving its spatial organization. The observations used to construct the state and the stages selected for downstream readout are separate.

We instantiate this view in **StateVid**, which augments a pretrained SigLIP2 encoder (Tschannen et al., 2025) with recurrent temporal blocks that maintain a persistent spatial state. At each spatial location, learned gates control how information from the current observation is incorporated and how previously accumulated content is retained. We implement these transitions with the Recurrent Rule (Yang et al., 2025; Dao & Gu, 2024; Peng et al., 2025). Unlike frame-wise temporal aggregation, the resulting state itself evolves through the video and can be read out at selected stages. This formulation naturally subsumes images. The snapshot schedule sets the visual token budget exposed downstream. Short captioned clips (Xu et al., 2016; Anne Hendricks et al., 2017; Bain et al., 2021; Wang et al., 2022) provide supervision for local visual content but offer limited signal for how information should persist or be discarded over time. To expose the model to state transitions beyond individual clips, we concatenate captioned clips into continuous streams and supervise retrieval throughout the resulting trajectories.

Across experiments, StateVid surpasses visual encoders trained on far larger datasets on zero-shot video retrieval and action recognition tasks, while maintaining image performance. Under a fixed downstream visual token budget, it can improve the average performance of VideoLLM and benefit more from additional observed frame inputs. Moreover, when sampling density increases over a fixed video duration, StateVid remains stable retrieval performance. State maps show spatially selective responses to visual changes. Our main contributions are as follows:

- We formulate video representation as an evolving visual state, treating the state trajectory rather than a sequence of frame-wise features as the video representation.
- We introduce StateVid, a recurrent spatial encoder with gated state updates and ordered snapshot readout, providing a unified image–video interface whose downstream representation budget is decoupled from the number of observed frames.
- We develop a short-clip training strategy that concatenates captioned clips into continuous streams to learn retention and revision without long-video annotations.

2 Related Work

Video Encoder and Representation. ViT represents an image as a sequence of patch tokens (Dosovitskiy et al., 2020). Video Transformers extend this representation along the temporal dimension, using frame-wise patches or tubelets and spatiotemporal interactions to encode video (Bertasius et al., 2021; Arnab et al., 2021). Recent video foundation models further acquire spatial and semantic knowledge from pretrained image models through feature distillation (Radford et al., 2021; Zhai et al., 2023; Tschannen et al., 2025; Li et al., 2023; Wang et al., 2024). Another line of work directly transfers pretrained image ViTs to video. Early methods aggregate independently encoded frame features (Fang et al., 2021; Ma et al., 2022; Lin et al., 2022), while later methods insert temporal modules into or alongside the image backbone (Pan et al., 2022; Yang et al., 2023; Liu et al., 2023; 2024b; Xue et al., 2023; Zhang et al., 2024a). Although these approaches retain strong image representations, they are typically trained on short clips with 8 to 32 frames and still model time over frame-wise patch tokens. Consequently, their token count grows with video length, and the resulting context becomes the bottleneck for the hundreds of frames processed by modern VideoLLMs. Stat-

eVid instead represents video as a recurrently evolving visual state and extends encoder pretraining to hundreds of frames by concatenating short clips.

Adapting visual features for VideoLLMs. As multimodal models unify single-image, multi-image, and video understanding, a shared visual backbone has become a common choice (Bai et al., 2025b;a; Zhang et al., 2023). However, frame-wise encoding makes visual tokens and LLM context grow with video length. Existing methods compress frame-wise features through resampling, hierarchical aggregation, sparse memory, or adaptive token selection (Song et al., 2024; He et al., 2024; Tang et al., 2025; Shen et al., 2024). Others combine clip-level spatiotemporal modeling and token merging with token dropout in the LLM (Jin et al., 2024; Li et al., 2026; Wang et al., 2025b). Training-free methods pool, prune, or merge visual tokens based on attention, approximation error, or inter-frame novelty (Luo et al., 2026; Tang et al., 2026; Ma et al., 2026). StateVid instead integrates frames during visual encoding through recurrent state updates, representing temporal changes with richer features rather than more tokens.

From Recurrent Token Mixing to State Representation. Linear attention admits an RNN-like formulation in which kernelized attention maintains a matrix-valued recurrent state updated online for each token, reducing the streaming cost from quadratic to linear in sequence length (Katharopoulos et al., 2020). Retention makes this recurrence explicit through decayed state updates (Sun et al., 2023), while gated delta rules combine selective forgetting with targeted writes (Yang et al., 2025). In vision, RMT applies retention across visual tokens (Fan et al., 2023), ViG and Vision RWKV use gated or recurrent token interactions (Liao et al., 2024; Duan et al., 2024), and Vision Mamba and VMamba use selective scans (Zhu et al., 2024; Liu et al., 2024c). Across these designs, including BIMBA Islam et al. (2025), the recurrent state primarily carries information used to mix, contextualize, or select token features. StateVid instead uses recurrence to construct the representation itself. Its spatially arranged state persists across observations, is supervised across concatenated clips, and is exposed through ordered snapshots for retrieval and VideoLLM conditioning. The distinction is representational rather than a claim of a new transition rule.

3 StateVid Architecture

StateVid interleaves spatial processing with temporal blocks that contextualize frame features and update a persistent visual state for snapshot readout. **It should be noted that StateVid can use any Recurrent Rule to construct temporal or state layers, and this section only uses one Rule.**

3.1 Overview

To extend the pretrained SigLIP2 (Tschannen et al., 2025) to video, we introduce temporal blocks that enable cross-frame interaction and recurrent state construction, as summarized in Fig. 2. The first L_0 spatial layers remain unchanged, while the remaining layers are interleaved with temporal blocks. Let $\mathcal{B}_l$ denote a spatial layer and $\mathcal{T}_k$ the corresponding temporal block. Given a video feature tensor $V \in \mathbb{R}^{T\times N\times D}$, where T is the number of frames, N the number of tokens per frame, and D the feature dimension, each temporal block groups tokens at the same spatial position across frames. It then processes these temporal sequences through two complementary paths. The Temporal Layer produces frame features enriched with cross-frame context, while the State Layer updates the recurrent visual state. Let $V^{(k)}$ be the input to temporal block $\mathcal{T}_k$. Its temporally contextualized features $\widetilde{V}^{(k)}$ are passed to the next spatial layer to produce $V^{(k+1)}$, whereas the State Layer updates $R^{(k-1)}$ and passes it to the next temporal block.

$$\begin{aligned} \widetilde{V}^{(k)} &= \mathrm{TemporalLayer}^{(k)}\left(V^{(k)}\right), \\ V^{(k+1)} &= \mathcal{B}_{L_0+k+1}\left(\widetilde{V}^{(k)}\right), \\ R^{(k)} &= \mathrm{StateLayer}^{(k)}\left(R^{(k-1)}, \widetilde{V}^{(k)}\right). \end{aligned} \tag{1}$$

3.2 Temporal Block and Visual State

StateVid use Recurrent Rule to construct frame features and evolve the state. In this section, we instantiate this Recurrent Rule with the Gated Delta Rule (GDR) (Yang et al., 2025) to give a example.

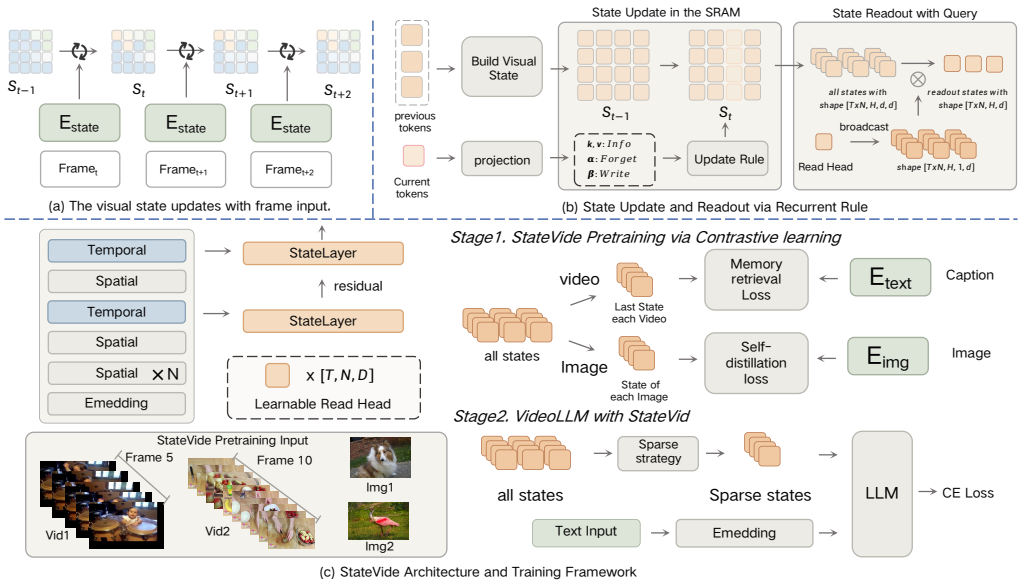

Figure 2: Overview of the StateVid architecture. Temporal blocks enrich frame features and update a recurrent spatial state, whose selected snapshots are exposed downstream.

Given a query q_t, a key k_t, and a value v_t, together with a decay gate $\alpha_t \in (0, 1)$ and a writing gate $\beta_t \in (0, 1)$, its update is:

$$S_t = S_{t-1}\left(\alpha_t\left(I - \beta_t k_t k_t^\top\right)\right) + \beta_t v_t k_t^\top, \qquad o_t = S_t q_t. \tag{2}$$

The decay gate controls retention, and the writing gate revises the association aligned with the current key before adding the new value.

Temporal Feature Construction. The Temporal Layer builds dynamic video features using information from preceding frames. Given $V^{(k)} \in \mathbb{R}^{T\times N\times D}$, we transpose it to $N \times T \times D$ and apply GDR independently to the temporal sequence at each spatial position. We compute:

$$q_{t,n}^{(k)} = W_q^{(k)} x_{t,n}^{(k)}, \quad (k_{t,n}^{(k)}, v_{t,n}^{(k)}) = W_{kv}^{(k)} x_{t,n}^{(k)}, \quad (\alpha_{t,n}^{(k)}, \beta_{t,n}^{(k)}) = \Phi_g^{(k)}\left(x_{t,n}^{(k)}\right). \tag{3}$$

GDR operates along the temporal dimension at each spatial position, so $\widetilde{V}^{(k)}$ retains the spatial organization of each frame while incorporating preceding-frame context.

Evolving State Construction. The State Layer maintains a visual state at each spatial position. At time t, it summarizes contextualized features up to t through a causal update.

$$R_{t,n} = \mathcal{U}\left(R_{t-1,n}, \widetilde{V}_{t,n}\right) = \mathcal{F}\left(\widetilde{V}_{1:t,n}\right). \tag{4}$$

The update is applied independently at each spatial position and is realized with GDR $S_{t,n}^{(k)}$.

$$S_{t,n}^{(k)} = \mathcal{U}_{\mathrm{GDR}}^{(k)}\left(S_{t-1,n}^{(k)}, \widetilde{V}_{t,n}^{(k)}\right) = \mathcal{F}_{\mathrm{GDR}}^{(k)}\left(\widetilde{V}_{1:t,n}^{(k)}\right). \tag{5}$$

The matrix is updated causally from the previous state and the current feature, so its size is independent of the number of frames. Let H_{head} denote the number of GDR heads, with per-head dimensions d_k and d_v. For head h at position n, the native state has shape $S_{t,n,h}^{(k)} \in \mathbb{R}^{d_v\times d_k}$, and all states together have shape $\mathbb{R}^{N\times H_{\mathrm{head}}\times d_v\times d_k}$. The update runs independently across positions and heads. Since the downstream representation has shape $H \times W \times D$, we map each matrix state to a compact head-wise feature with the GDR query operation, concatenate the heads, and project the result while preserving the patch layout. The first State Layer uses a learned readout query, and later layers derive it from the preceding explicit state.

$$R_{t,n,h}^{(k)} = S_{t,n,h}^{(k)} q_{t,n,h}^{R,(k)}, \qquad q_{t,n,h}^{R,(k)} = W_{q,h}^{R,(k)}\,\mathrm{LN}\left(R_{t,n}^{(k-1)}\right), \qquad R_{t,n}^{(0)} = r_0, \tag{6}$$

where $q_{t,n,h}^{R,(k)} \in \mathbb{R}^{d_k}$ maps the native matrix state to the head-wise readout $R_{t,n,h}^{(k)} \in \mathbb{R}^{d_v}$. Concatenated head readouts are projected and arranged on the patch grid to form the explicit state $R \in \mathbb{R}^{H \times W \times D}$ exposed for snapshot selection and downstream conditioning. Thus, S is the internal recurrent matrix memory and R is the patch-structured state passed downstream.

3.3 Sparse Snapshot Readout for Downstream Models

The State Layer l^* connected to the language model produces a trajectory $\{R_t^{(l^*)}\}_{t=1}^{T}$, where each state summarizes the video prefix up to time t. Since later updates can overwrite earlier content, a final-state-only readout can discard events from earlier in the video.

We retain K intermediate snapshots by partitioning the trajectory into consecutive stages and selecting their ordered endpoints $1 \le \tau_1 < \cdots < \tau_K \le T$. Each selected state is projected to the language model dimension, and the resulting tokens are concatenated in temporal order.

$$Z_{\text{video}} = \left[\text{Proj}\left(R_{\tau_1}^{(l^*)}\right); \ldots; \text{Proj}\left(R_{\tau_K}^{(l^*)}\right)\right]. \tag{7}$$

Each selected snapshot undergoes 2×2 spatial downsampling for VideoLLM input, leaving one quarter of its patch tokens. The stage length is adjusted to meet the snapshot budget, preserving intermediate states before overwrite. The number of visual tokens is therefore set by the readout schedule K and the spatial grid size rather than the frame count T. An image is treated as a single-step static video, so its projected image feature directly serves as Z_{image}. We denote either representation by Z_{visual} and insert it at the visual placeholder between the language model embeddings of the preceding and following text.

4 StateVid Training Recipe

4.1 Mixed Image-Video Training

StateVid is trained with alternating image and video batches. We treat each image as a static sequence by repeating it across several time steps and use the frozen SigLIP2 as the image teacher.

Let F_l and $\bar{F}_l$ denote the student and teacher image tokens at layer l, respectively. Over a set of selected layers $\mathcal{K}$, $\mathcal{L}_{\cos}$ is the mean token-wise cosine distance, *i.e.*, one minus the cosine similarity between corresponding student and teacher tokens. To further preserve the spatial structure among visual tokens, let $\mathcal{R}(F_l)$ denote the distribution of pairwise token similarities normalized with temperature 0.07 for both student and teacher. We align these relation distributions using

$$\mathcal{L}_{\text{rel}} = \frac{1}{|\mathcal{K}|} \sum_{l \in \mathcal{K}} D_{\text{KL}}\big(\mathcal{R}(\bar{F}_l) \,\|\, \mathcal{R}(F_l)\big). \tag{8}$$

We also match the image-text retrieval distributions produced by the student and teacher, both using temperature 0.1. Denoting the corresponding image-to-text and text-to-image distributions by P_{i2t}, P_{t2i}, $\bar{P}_{\text{i2t}}$, and $\bar{P}_{\text{t2i}}$, this objective is

$$\mathcal{L}_{\text{img-ret}} = \frac{1}{2}\left[D_{\text{KL}}\big(\bar{P}_{\text{i2t}} \,\|\, P_{\text{i2t}}\big) + D_{\text{KL}}\big(\bar{P}_{\text{t2i}} \,\|\, P_{\text{t2i}}\big)\right]. \tag{9}$$

The complete image training objective is

$$\mathcal{L}_{\text{img}} = \lambda_{\cos}\mathcal{L}_{\cos} + \lambda_{\text{rel}}\mathcal{L}_{\text{rel}} + \lambda_{\text{img-ret}}\mathcal{L}_{\text{img-ret}}. \tag{10}$$

4.2 Training State Updates with Concatenated Short Clips

The state needs supervision over successive updates, whereas most available training videos contain only short temporal contexts. We therefore concatenate the B clips within each training batch along time.

$$V^{\text{concat}} = \left[V^{(1)}; V^{(2)}; \ldots; V^{(B)}\right]. \tag{11}$$

The resulting sequence is processed without resetting the recurrent state. We retain the clip boundaries $\{b_i\}$ to read the state associated with each clip and apply clip-level supervision. Because

adjacent clips are generally unrelated, the state carried from the preceding clip introduces a strong historical prior at each boundary. Instead of clearing this state, we strengthen the decay at the first step of every new clip. Let $g_t < 0$ be the log-decay used to produce the GDR retention gate $\alpha_t = \exp(g_t)$. At a clip boundary, we apply:

$$g_{b_i} \leftarrow \kappa g_{b_i}, \qquad \alpha_{b_i} \leftarrow \alpha_{b_i}^{\kappa}, \qquad \kappa > 1. \tag{12}$$

This boundary-enhanced decay with $\kappa = 3$ suppresses the strong prior from the preceding clip while preserving continuous state transitions. The concatenated sequence provides multiple state updates with supervision at each clip endpoint. Boundary-enhanced decay is applied only while building training streams. At inference a video is one continuous sequence, so the state is updated from its first frame to its last and no boundary is inserted.

4.3 Causally Weighted Video-Text Retrieval

For the concatenated sequence, the state at the end of clip i contains its current content together with a progressively compressed trace of earlier clips. We supervise this property with a causally weighted video-text retrieval objective. For normalized endpoint-state and caption embeddings $\mathbf{v}_i$ and $\mathbf{t}_j$, the logit is fixed at $\ell_{ij} = 100\,\mathbf{v}_i^\top \mathbf{t}_j - 15$; captions in the current batch and a queue of text negatives are included. The current caption receives full positive weight, while captions from preceding clips remain positive with exponentially decaying weights according to their temporal distance. Future captions and queued texts are treated as negatives. Let y_{ij} denote the target weight. We use a sigmoid loss rather than a standard contrastive objective. The resulting video-to-text retrieval loss is

$$\mathcal{L}_{\text{ret}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{B+Q}\left[y_{ij}\log\sigma(\ell_{ij}) + (1-y_{ij})\log(1-\sigma(\ell_{ij}))\right], \tag{13}$$

where Q is the number of valid texts in the queue. The decayed historical positives encourage the recurrent state to retain earlier content, while the boundary-enhanced decay in Eq. 12 prevents irrelevant history from dominating the current clip. Together, they supervise how the state updates as new clips arrive.

5 Experiments

5.1 Experiment Setting

Implementation Details. Encoder pretraining. StateVid interleaves temporal blocks among the last four spatial layers of SigLIP2. Pretraining uses 2M videos from InternVid-10M (Wang et al., 2022), recaptioned with Qwen3-VL-4B, and images from CC12M (Changpinyo et al., 2021), ImageNet-21K, and their recaptioned versions. We sample at most 24 frames per video clip and concatenate the 20 clips in each video batch. The encoder is trained for 20K steps with alternating image and video batches at a learning rate of 5×10^{-4}, with a negative queue of Q=8192 and a decay factor of ρ=0.2.

Tasks and Benchmarks. We evaluate zero-shot retrieval on MSR-VTT (Xu et al., 2016), DiDeMo (Anne Hendricks et al., 2017), and ActivityNet (Krishna et al., 2017), and action recognition on Kinetics-400 (Kay et al., 2017), UCF101 (Soomro et al., 2012), and HMDB51 (Kuehne et al., 2011). Image capability is evaluated on ImageNet-1K (Deng et al., 2009), CIFAR-10/100 (Krizhevsky, 2009), Caltech101 (Li et al., 2004), VOC2007 (Everingham et al., 2010), Stanford Cars (Krause et al., 2013), Oxford-IIIT Pets (Parkhi et al., 2012), Food-101 (Bossard et al., 2014), Oxford Flowers (Nilsback & Zisserman, 2008), EuroSAT (Helber et al., 2019), and RESISC45 (Cheng et al., 2017).

5.2 Encoder Performance

Strong Zero-shot Video Task Performance. Tab. 1 shows that StateVid improves both SigLIP2 initializations across all six retrieval metrics and three action-recognition benchmarks. Trained on 2M videos, StateVid-Base surpasses the larger PE_{core}G on DiDeMo T2V. StateVid-So400M exceeds PE_{core}G on four of six retrieval metrics and HMDB51, and outperforms InternVideo2-1B on

Table 1: Zero-shot video encoder comparison on retrieval (R@1) and action recognition. Bold and underline denote the best and second-best result within each parameter group.

Video Encoder	Video Data	MSR-VTT		DiDeMo		ActivityNet		Action Recognition		
		T2V	V2T	T2V	V2T	T2V	V2T	K400	UCF101	HMDB51
≥ 1B parameters										
VideoCoCa (Yan et al., 2022)	100M	34.3	64.7	–	–	34.5	33.0	72.0	86.6	58.7
InternVideo2-1B (Wang et al., 2024)	102M	51.9	50.9	57.0	54.3	60.4	54.8	73.2	88.3	53.9
PE_{core}G (Bolya et al., 2026)	22M	51.2	49.9	44.9	45.8	54.7	51.2	76.9	90.7	61.1
≤0.1B parameters										
UMT-B (Li et al., 2023)	25M	35.2	30.3	41.2	40.8	35.5	32.8	–	–	–
VideoMamba (Li et al., 2024b)	25M	35.6	–	43.1	–	41.0	–	–	–	–
SigLIP2-B/16 (Tschannen et al., 2025)	–	38.6	28.9	32.7	28.7	25.3	24.2	57.3	78.8	42.4
PE_{core}B (Bolya et al., 2026)	22M	**47.6**	47.3	36.5	35.2	39.0	38.4	**65.6**	**84.6**	48.2
StateVid-Mamba2-Base	2M	46.2	46.8	44.6	42.5	43.1	40.0	65.1	82.0	**53.3**
StateVid-GDR-Base	2M	46.8	**47.5**	**45.7**	**42.9**	**43.4**	**40.4**	64.9	82.1	52.3
≤0.5B parameters										
InternVideo-L (Wang et al., 2022)	12.8M	40.7	39.6	31.5	33.5	30.7	31.4	64.2	–	–
UMT-L (Li et al., 2023)	25M	40.7	37.1	48.6	**49.9**	41.9	39.4	–	–	–
BT-Adapter (Liu et al., 2024b)	2M	40.9	–	35.6	–	37.0	–	64.3	79.1	54.6
SigLIP2-So400M (Tschannen et al., 2025)	–	42.8	31.0	38.5	31.2	34.2	30.3	67.7	87.7	52.4
PE_{core}L (Bolya et al., 2026)	22M	50.3	50.1	40.2	39.7	46.4	42.1	**73.4**	87.1	58.5
StateVid-So400M	2M	**51.9**	**51.0**	**49.8**	49.1	**52.0**	**49.3**	73.1	**89.8**	**62.5**

Table 2: Zero-shot image classification across 11 benchmarks. Bold and underline denote the best and second-best result within each group.

Model	ImageNet 1K	CIFAR 10	Caltech 101	CIFAR 100	VOC 2007	Stanford Cars	Oxford Pets	Food 101	Oxford Flowers	EuroSAT	RESISC45	Avg.
SigLIP2 Base-224 (Tschannen et al., 2025)	78.2	93.7	86.9	74.3	81.7	93.4	95.4	92.5	85.9	48.0	70.9	81.9
StateVid Base	78.0	93.4	**87.2**	74.3	81.6	92.7	94.7	92.0	85.7	48.1	71.7	81.8 (−0.1)
UMT-L16 (Li et al., 2023) w/ rerank	40.8	92.0	61.3	67.4	67.1	12.9	40.1	39.7	21.3	25.8	40.5	46.3
InternVideo2-1B (Wang et al., 2024) w/ rerank	34.6	96.6	72.6	74.4	75.1	12.9	33.1	46.6	20.8	51.0	40.0	50.7
CLIP ViT-L14-336 (Radford et al., 2021)	76.5	95.0	82.1	77.1	77.3	79.3	93.8	93.8	78.3	**63.0**	71.4	80.7
BTAdapter-L14 (Liu et al., 2024b)	72.0	94.4	82.2	69.6	77.7	69.9	92.0	90.7	72.4	57.2	67.8	76.9 (−3.8)
SigLIP2 So400m-384 (Tschannen et al., 2025)	**84.1**	**96.9**	84.7	**82.7**	**82.1**	**95.8**	**96.6**	**96.7**	**89.2**	52.8	75.0	**85.2**
StateVid-So400m	83.9	96.3	84.8	82.2	81.8	95.6	96.5	96.5	88.7	52.6	**76.0**	85.0 (−0.2)

UCF101 and HMDB51. Within its parameter group, StateVid-So400M also surpasses PE_{core}L on all six retrieval metrics and achieves the strongest UCF101 and HMDB51 results. These comparisons show that the learned representation transfers across retrieval and action recognition, including settings with substantially larger competing encoders. The SigLIP2 baselines and the PE_{core} DiDeMo results are rerun with our evaluation code; other external entries are taken from their cited papers.

Maintain Image Capability. By decoupling temporal modeling from spatial processing, StateVid accommodates images as static videos within the same framework. Tab. 2 shows minimal loss in image recognition accuracy after video training, supporting our design for extending image encoders with temporal capabilities while preserving their image competence. StateVid-Base and StateVid-So400M remain within only 0.1 and 0.2 percentage points of their corresponding image-only backbones, respectively. This small degradation indicates that the temporal modules and mixed image–video training largely preserve the pretrained spatial and semantic representation.

Effect of Concatenated Clips and State Modeling. StateVid training has two main components, concatenating short clips into streams and the state layer, and we ablate both. The second and last rows of Tab 3 isolate the effect of concatenation, that MSR-VTT changes from 47.5% to 47.2% R@1, while DiDeMo improves from 43.0% to 44.1% and ActivityNet from 40.9% to 41.9%. These results show a modest benefit from supervising successive state updates. The third row is a Temporal-GDR-only baseline: it retains the temporal path and clip concatenation, removes the State Layer, and mean-pools the temporally contextualized features. Comparing it with the last row isolates the contribution of the State Layer over mean readout. The mean score increases by 3.0% from 41.4% to 44.4%, with gains of 0.3% on MSR-VTT, 3.9% on DiDeMo, and 4.7% on ActivityNet.

State Transitions under Denser Observation. Tab. 3 shows that StateVid improves the three-benchmark average by 2.2% over attention aggregation. Fig. 3 keeps each video's temporal span fixed while increasing the number of uniformly sampled frames to 96. StateVid's retrieval performance remains relatively stable, while attention aggregation declines at higher frame counts, so their gap generally widens with denser sampling. The widening is clearest on ActivityNet, where the V2T R@1 gap grows from nearly zero at 16 frames to about 4.9% at 96 frames, and the T2V gap grows

Table 3: Ablation of clip concatenation and temporal state construction. State versus mean pooling and attention.

Clip Concat.	State Layer	Temporal Aggregation	MSR	DiDe	ANet	Avg.
×	×	Mean	33.8	30.7	24.8	29.7
×	✓	State	**47.5**	43.0	40.9	43.8
✓	×	Mean	46.9	40.2	37.2	41.4
✓	✓	Attn	46.2	41.0	39.6	42.2
✓	✓	State	47.2	**44.1**	**41.9**	**44.4**

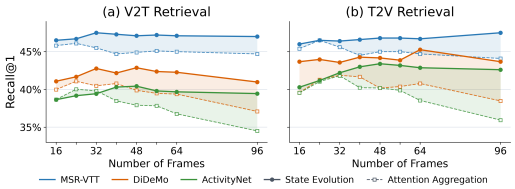

Figure 3: Dense-observation robustness of state and attention aggregation with up to 96 frames sampled over the same video duration.

to 6.7%. This pattern is consistent with the repeated-observation analysis in App. E. And we provide more Recurrent Rule results in App. C, which show the effective of video state.

5.3 VideoLLM Tuning

We train the VideoLLM in three stages. Stage 1 warms up the projector on LLaVA-ReCap-558K (Liu et al., 2024a) at a learning rate of 1×10^{-3}. Stage 2 aligns short videos of up to 32 frames on ∼0.3M videos mixed with 0.2M images (Zhang et al., 2026; An et al., 2026; Yuan et al., 2025). Stage 3 conducts long-video and instruction tuning on ∼0.4M videos mixed with 0.2M images (Farré et al., 2024; Liu et al., 2024a; Zhang et al., 2024b). Stages 2–3 jointly tune the LLM, projector, and temporal layers at 2×10^{-5} with a global batch size of 64 for one epoch. We evaluate on LongVideoBench (LVB) (Wu et al., 2024), LVBench (LV-B) (Wang et al., 2025a), MLVU (Zhou et al., 2025), PerceptionTest (PT) (Patraucean et al., 2023), and VideoMME (Fu et al., 2025).

Protocol-matched VideoLLM comparison. Let T denote observed frames and K the visual readouts passed to the LLM. Tab. 4 fixes K=32 frame token budget for all methods. Mean reflects temporal pooling used in recent video systems (Luo et al., 2026), Sim mean reflects similarity-aware redundancy handling (Shen et al., 2024; Song et al., 2024), and ToMe reflects the token merging used by VideoChat-Flash (Li et al., 2026). StateVid improves the SigLIP2 average by 1.5 points and leads on four of five benchmarks, with Sim mean highest on VideoMME. Under this shared downstream recipe, the comparison measures how each visual representation is used by the same VideoLLM rather than standalone vision quality. The result supports constructing a video representation through state updates before sparse readout.

Video representation at a fixed downstream budget. Panel a of Fig. 4 fixes K=32 and increases T from 32 to 256. StateVid integrates each observation into its recurrent spatial state before selecting the same number of ordered snapshots for the LLM. Its plotted score rises from about 43.5 to 45.1 at 128 frames and remains about 1.2 points above Mean at 256 frames, while Mean declines from about 44.0 to 43.1. The result shows how the state trajectory can absorb additional observations without increasing the downstream visual-token budget.

Effect of the readout budget. Panel b of Fig. 4 increases the readout budget from K=8 to 64 under the plotted sampling protocols. StateVid leads both baselines across this range and reaches the highest score at K=64. The two panels separate the benefit of integrating more observations into the persistent state from the benefit of exposing more snapshots of that state to the LLM. Additional budget and retention probes are reported in App. D.4 and App. D.

5.4 Visualization Analyse

Historical supervision shapes gate responses. Shown as Fig. 5 compares historical-target decay ρ with Cut and No forget on concatenated clips. At $\rho = 0$, writing is weak and forgetting is relatively high; Cut writes strongly at boundaries without content-dependent forgetting. With $\rho = 0.2$, retrieval improves over both controls, and writing and forgetting rise near transitions before settling within clips. Training-only boundary-enhanced decay is disabled for visualization, so these responses are learned. Historical supervision thus balances retention and revision; these choices also affect dense-observation robustness and interference tolerance (App. D).

Table 4: VideoLLM comparison with shared tuning data and visual-token budget at K=32. Δ is the average-point change from SigLIP2.

Encoder	LVB	LV-B	MLVU	PT	Video MME	Avg.	Δ
SigLIP2	48.3	36.4	51.6	50.7	50.8	47.6	–
+ Mean	44.8	36.2	51.3	49.6	50.6	46.5	−1.1
+ Sim mean	46.8	37.4	51.4	50.1	**53.2**	47.8	+0.2
+ ToMe	47.1	37.1	51.7	50.6	50.2	47.4	−0.2
StateVid	**49.3**	**38.9**	**54.4**	**50.8**	52.0	**49.1**	**+1.5**

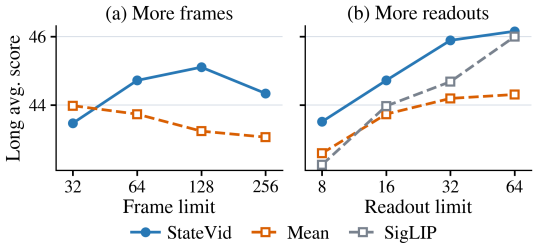

Figure 4: VideoLLM observation and readout budgets. (a) Increasing T at K=32. (b) Increasing K for StateVid and Mean.

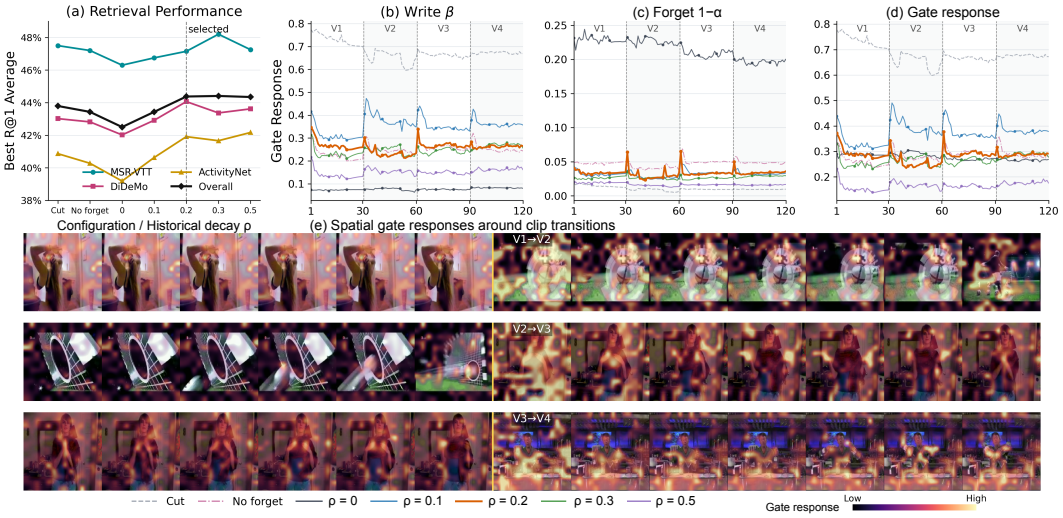

Figure 5: Gate analysis across historical-target decay settings ρ. (a) Retrieval performance. (b–d) Write, forgetting, and combined gate responses. (e) Spatial gate responses around clip transitions.

Spatially selective gate responses. A next generation video encoder should automatically focus on content that changes over time, much like a video codec allocates more information to visually changing regions (An et al., 2026). Figure 5 shows the learned responses around three clip transitions. Strong responses concentrate on changing people, hands, and objects involved in interactions, while static background regions receive weaker responses. The active regions vary across transitions and do not follow a fixed global reset pattern. These observations show that StateVid exhibits the desired content adaptive behavior. Its learned state updates focus on locations where new visual evidence appears while preserving less affected regions.

6 Conclusion

StateVid reframes video representation learning around the evolution of a persistent visual state. Its central contribution is to separate the acquisition of visual evidence from the representation exposed downstream. Temporal information is encoded through updates to spatial features, allowing additional observations to inform a bounded representation whose readout size is independent of the number of observations while preserving an image-compatible interface. Our training method extends state supervision across clip boundaries using short captioned videos. The empirical results support the value of this formulation for video understanding while largely retaining the image capabilities of the pretrained encoder. The VideoLLM study evaluates this interface under a shared downstream recipe. Together, these findings motivate a direction for video representation learning centered on what a state retains and revises as observations accumulate, with observation and readout budgets treated as independent design choices.

REFERENCES

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, et al. Llava-onevision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pp. 5803–5812, 2017.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 6816–6826. IEEE, 2021.

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, 2025a. URL https://arxiv.org/abs/2511.21631.

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, 2025b. URL https://arxiv.org/abs/2502.13923.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Bangalath, et al. Perception encoder: The best visual embeddings are not at the output of the network. *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality, 2024. URL https://arxiv.org/abs/2405.21060.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yuchen Duan, Weiyun Wang, Zhe Chen, Xizhou Zhu, Lewei Lu, Tong Lu, Yu Qiao, Hongsheng Li, Jifeng Dai, and Wenhai Wang. Vision-RWKV: Efficient and scalable visual perception with RWKV-like architectures. *arXiv preprint arXiv:2403.02308*, 2024.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010. doi: 10.1007/s11263-009-0275-4.

Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. Rmt: Retentive networks meet vision transformers. *arXiv preprint arXiv:2309.11523*, 2023.

Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. `https://huggingface.co/datasets/HuggingFaceFV/finevideo`, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, June 2025. URL `https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html`.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13504–13514, 2024. doi: 10.1109/CVPR52733.2024.01282.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29096–29107, June 2025.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13700–13710. IEEE, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5156–5165, 2020.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 706–715, 2017. doi: 10.1109/ICCV.2017.83.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso A. Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2556–2563, 2011. doi: 10.1109/ICCV.2011.6126543.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Fei-Fei Li, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 178–178, 2004.

Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19891–19903. IEEE, 2023.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, 2024b. URL https://arxiv.org/abs/2403.06977.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. In *International Conference on Learning Representations*, volume 2026, pp. 109089–109117, 2026.

Bencheng Liao, Xinggang Wang, Lianghui Zhu, Qian Zhang, and Chang Huang. Vig: Linear-complexity visual sequence learning with gated linear attention. *arXiv preprint arXiv:2405.18425*, 2024.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pp. 388–404. Springer, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024a.

Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6555–6564. IEEE, 2023.

Ruyang Liu, Chen Li, Yixiao Ge, Thomas H Li, Ying Shan, and Ge Li. Bt-adapter: Video conversation is feasible without video instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13658–13667. IEEE, 2024b.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024c.

Bingjun Luo, Tony Wang, Hanqi Chen, and Xinpeng Ding. Enhancing visual token representations for video large language models via training-free spatial-temporal pooling and gridding. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MZi9SYPVz5.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. Apet: Approximation-error guided token compression for efficient vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26306–26316, 2026.

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 638–647, 2022.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, 2024.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and Joao Carreira. Perception Test: A diagnostic benchmark for multimodal video models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42748–42761, 2023. doi: 10.52202/075280-1852. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8540fba4abdc7f9f7a7b1cc6cd60e409-Abstract-Datasets_and_Benchmarks.html.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, Guangyu Song, Kaifeng Tan, Saiteja Utpala, Nathan Wilce, Johan S. Wind, Tianyi Wu, Daniel Wuttke, and Christian Zhou-Zheng. Rwkv-7 "goose" with expressive dynamic state evolution, 2025. URL https://arxiv.org/abs/2503.14456.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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. *arXiv preprint arXiv:2410.17434*, 2024.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232. IEEE, 2024.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *arXiv preprint arXiv:1212.0402*, 2012.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Chong Tang, Sannara Ek, Dirk Koch, Robert Mullins, Alex Weddell, and Jagmohan Chauhan. Surge: Surprise-guided token reduction for efficient video understanding with vlms. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rCFPSIjqMf.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29118–29128. IEEE, 2025.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967, October 2025a. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_LVBench_An_Extreme_Long_Video_Understanding_Benchmark_ICCV_2025_paper.html.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European conference on computer vision*, pp. 396–416. Springer, 2024.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2.5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025b.

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*, volume 37, pp. 28828–28857, 2024. doi: 10.52202/079017-0907. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/329ad516cf7a6ac306f29882e9c77558-Abstract-Datasets_and_Benchmarks_Track.html.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment, 2023. URL https://arxiv.org/abs/2209.06430.

Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition, 2023. URL https://arxiv.org/abs/2302.03024.

Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

Boqiang Zhang, Lei Ke, Ruihan Yang, Qi Gao, Tianyuan Qu, Rossell Chen, Dong Yu, et al. Penguin-vl: Exploring the efficiency limits of vlm with llm-based vision encoders. *arXiv preprint arXiv:2603.06569*, 2026.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pp. 543–553, 2023.

Wei Zhang, Chaoqun Wan, Tongliang Liu, Xinmei Tian, Xu Shen, and Jieping Ye. Enhanced motion-text alignment for image-to-video transfer learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18504–18515. IEEE, 2024a.

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*, 2024b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhou_MLVU_Benchmarking_Multi-task_Long_Video_Understanding_CVPR_2025_paper.html.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2401.09417.

A AI Assistance Claim

In this work, we used generative AI tools to create or modify scientific figures or images, create or edit software code, assist with parts of mathematical proofs, and edit the research paper to improve readability. We have not used generative AI tools for tasks requiring disclosure beyond these uses, and other disclosure requirements are not applicable to this work. We have reviewed and verified all AI-assisted work, including manually checking AI-assisted modifications to scientific figures and images, reviewing and testing AI-assisted code for correctness, carefully verifying the mathematical reasoning and proofs with the authors' own derivations, and reviewing all AI-assisted edits to the manuscript for accuracy, clarity, and consistency. We take full responsibility for the final content of this work, including all text, mathematical claims and proofs, figures, images, and software code produced with the aid of generative AI.

B Discussion of Video Representation and Sparse Readout

Prior work has already explored recurrent and selective-scan mechanisms for visual sequence compression. In particular, BIMBA (Islam et al., 2025) constructs a shorter set of visual queries from many video observations and passes the selectively scanned query outputs to the LLM. Therefore, decoupling the observation budget from the number of emitted visual queries is not unique to StateVid. Likewise, the Gated Delta Rule used in our main model is adopted from prior work (Yang et al., 2025) and is not claimed as a new transition rule.

Our distinction is the organization and supervision of the memory. StateVid maintains a causal recurrent memory independently at each spatial position inside the visual encoder; updates this memory for every observation; trains persistence and revision across concatenated captioned clips using endpoint retrieval targets; and exposes ordered snapshots of the evolving spatial state as the common image–video interface. In contrast, the recurrent state in the cited token-mixing or query-selection methods is used to contextualize or select their output features. The claim is thus a combination of spatial state organization, cross-clip history supervision, and ordered state-snapshot readout, rather than exclusive ownership of input–output budget decoupling.

Our main experiments use the Gated Delta Rule (Yang et al., 2025) as the recurrent rule. In the supplementary material, we additionally report video retrieval results using Mamba2 (Dao & Gu, 2024), RetNet (Sun et al., 2023), and RWKV7 (Peng et al., 2025) as alternative recurrent rules. As the sampling density increases, all of these recurrent variants show better retrieval performance than architecturally matched models using standard attention aggregation. This result suggests that using recurrent rules to collect and process historical information can be more effective than directly querying a frame-wise KV cache.

Although our experiments provide evidence that state-based representation is effective and supports a stable unified image–video interface, they have limitations. Due to resource constraints, our experiments are based on SigLIP2-scale models, and we do not conduct large-scale, high-quality VideoLLM fine-tuning. The performance ceiling of this representation at larger model and training scales therefore remains an open question.

C Effect of Different Recurrent Rules

StateVid does not require a particular recurrent rule to construct its visual state. We therefore compare alternative recurrent rules while keeping the visual backbone, training data, optimization schedule, and state readout fixed. The table reports retrieval at 96 uniformly sampled frames over a fixed video duration, while the accompanying curves show how retrieval changes as the sampling density increases. These experiments test whether the state representation remains effective when the recurrent transition is changed and when the model receives denser observations.

Mamba2 and RWKV7 obtain performance close to GDR across the evaluated retrieval settings, whereas RetNet is less competitive. RWKV7 is overall stronger than the GDR configuration used in the main paper, showing that the recurrent transition itself is not the sole source of the observed gains. More importantly, the recurrent variants exhibit similar robustness as the observation density increases, outperforming the frame-wise attention aggregation baseline in the dense-observation

regime. These results support the broader conclusion that constructing and reading out a persistent visual state provides effective temporal modeling and favorable extrapolation to denser observations, while leaving room for further improvements from the choice of recurrent rule.

D Additional Ablations on Selective State Management

The following experiments examine state updates, retention under visual interference, and the effect of observation and readout budgets. In the encoder plots, the legend variable α denotes the historical-target decay ρ in Eq. 21. The GDR retention gate is α_t in Eq. 2. *Cut* resets the state at training clip boundaries. *No forget* retains the learned GDR gates and removes boundary-enhanced decay.

D.1 Dense Observation and Gate Responses

Retrieval with denser observation. Figure 7 varies sampling density over a fixed video duration. With $\rho = 0.2$, V2T R@1 changes from 46.5% to 47.0% on MSR-VTT and from 38.6% to 39.5% on ActivityNet as the frame count rises from 16 to 96. Retrieval remains broadly stable across the three datasets. The No-forget control declines on DiDeMo and ActivityNet.

Gate responses at clip boundaries. In Fig. 5, the ratio of forgetting-gate responses at training clip boundaries to responses within clips rises from 1.00 at $\rho = 0$ to 1.34 at $\rho = 0.1$ and 1.80 at $\rho = 0.2$. Boundary-enhanced decay is used only while training on concatenated clips. The visualized gates are collected with this intervention disabled, so the transition responses are produced by the learned model rather than imposed at inference.

D.2 Retention under Visual Interference

Retention through blank observations. We append up to 64 white frames after 32 video frames and retrieve from the final readout. In Fig. 8, StateVid with $\rho = 0.2$ retains about 36% V2T and 46% T2V R@1 after 64 blanks, compared with about 18% and 40% for standard attention. The comparisons below also include $\rho = 0.5$ and natural distractor frames.

Figure 9 compares both state settings with standard attention after the same suffix. At 64 blank frames, $\rho = 0.5$ reaches about 41% V2T and 45% T2V R@1, versus about 36% and 46% for $\rho = 0.2$. Both remain above standard attention, at about 18% and 40%; the larger historical-target weight improves V2T retention but is slightly lower on T2V.

Retention under natural interference. Figure 10 replaces the blank suffix with frames from another video. Target-pair retrieval declines as the distractor grows, but both state settings remain above standard attention. At 64 distractor frames, $\rho = 0.5$ retains about 23% V2T and 25% T2V R@1, compared with about 13% and 14% for $\rho = 0.2$ and about 1% and 2% for attention. The larger historical-target weight is more effective under long natural interference, while Fig. 9 shows a smaller, direction-dependent difference under blank-frame interference.

D.3 Writing and Retaining New Evidence

Sensitivity to a new observation. Figure 11 inserts a rendered text page among 32 video frames and measures retrieval at the page and at the sequence end. When the page appears last, final R@1 rises from 11.9% at $\rho = 0$ to 26.0% at $\rho = 0.2$. No forget retains the inserted text more strongly at this position, while its blank-frame retrieval falls more sharply in Fig. 8. These controlled probes show how historical supervision affects both retention of earlier video content and writing of new evidence.

D.4 Observation and Readout Budget Sweep

Tables 5–7 report the four-benchmark sweep. T caps observed frames and K caps ordered readouts passed to the LLM. Avg. is the unweighted mean of LongVideoBench, LVBench, MLVU, and full VideoMME without subtitles. For StateVid and Mean, the observation-to-readout ratio is T/K. The shared 2×2 spatial downsampling makes the input-to-output visual-token ratio $4T/K$.

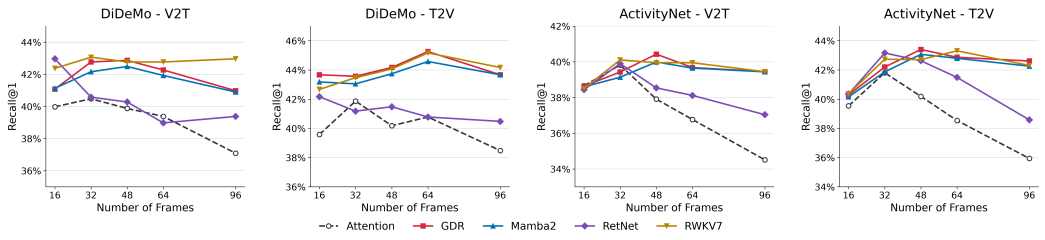

Figure 6: Dense-observation robustness of state and attention aggregation with up to 96 frames sampled over the same video duration.

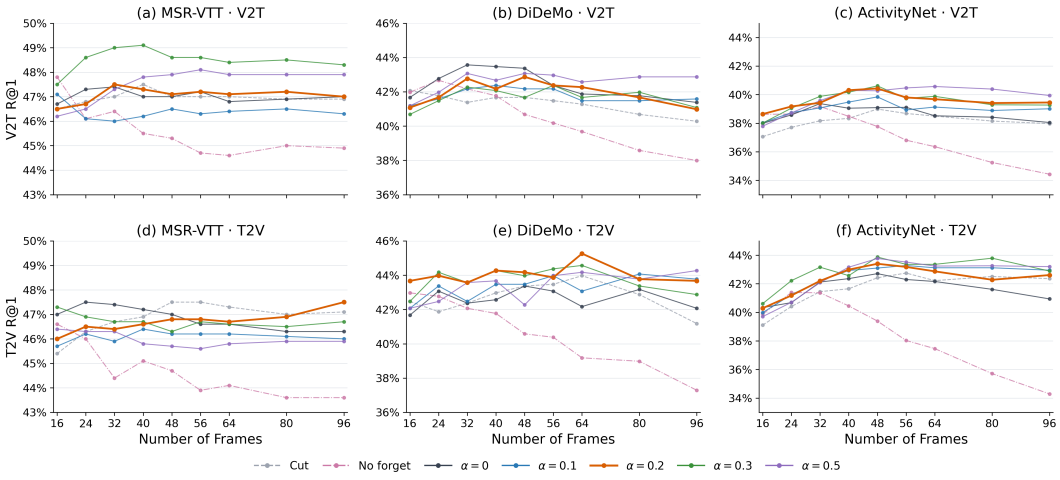

Figure 7: Dense-observation robustness in zero-shot retrieval as the number of uniformly sampled frames increases from 16 to 96 over the same video duration. The legend uses α for historical-target decay ρ.

At fixed $K = 32$, raising T from 32 to 256 increases StateVid's Avg. from 44.58 to 48.52, while Mean changes from 47.18 to 45.66. Additional observations improve the state representation with the same number of LLM readouts. With one readout per eight observed frames, StateVid scores 48.52, 48.59, and 48.52 at $T = 256$, 512, and 1,024, respectively. The nearly unchanged scores characterize stability beyond the 256-frame VideoLLM tuning cap and the 480-frame encoder training streams, rather than continued gains with video length. At $K = 64$, StateVid at $T = 512$ scores 48.59, while frame-wise SigLIP at $T = 64$ scores 48.70. Across the sweep, the advantage is concentrated in selected readout budgets and sparse observation-to-readout ratios, rather than being universal. These results measure stability under larger observation caps, while shorter videos contribute fewer frames than the configured cap.

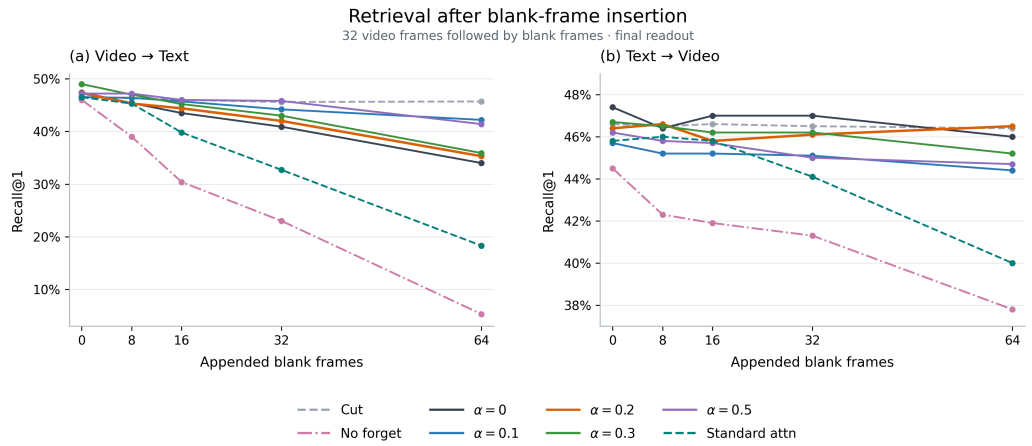

Figure 8: Robustness to an uninformative suffix. Each example contains 32 video frames followed by up to 64 identical blank frames. Retrieval uses the final readout. The legend uses α for historical-target decay ρ.

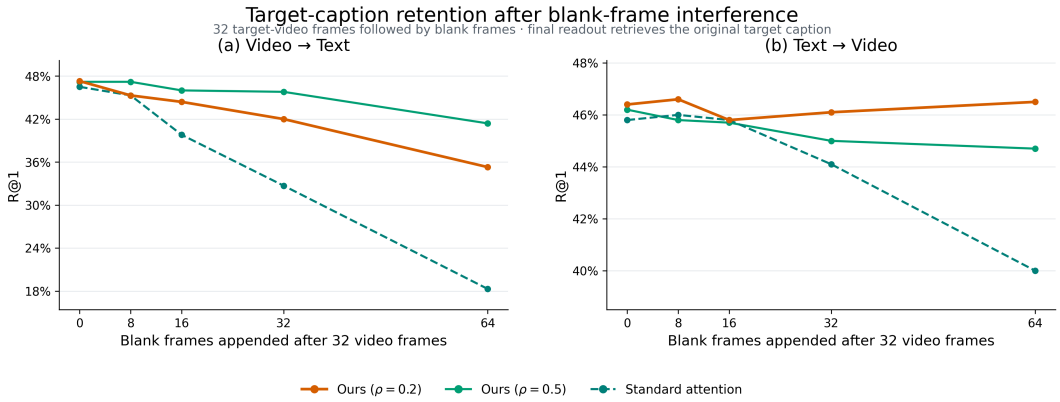

Figure 9: Target-pair retrieval after 32 video frames followed by 0–64 blank frames. The final representation is evaluated for StateVid with $\rho = 0.2$ or 0.5 and for standard attention.

T	K	Obs./readout	Token ratio	LongVideoBench	LV-Bench	MLVU	VideoMME	Avg.
32	8	4×	16×	47.05	36.48	50.53	49.93	45.99
32	16	2×	8×	47.79	36.15	50.60	50.26	46.20
32	32	1×	4×	47.49	34.60	48.39	47.81	44.58
64	8	8×	32×	46.45	36.02	52.32	49.89	46.17
64	16	4×	16×	47.64	36.93	53.74	50.52	47.21
64	32	2×	8×	46.67	37.38	51.98	50.63	46.67
64	64	1×	4×	45.70	35.51	50.52	47.33	44.76
128	8	16×	64×	45.77	35.12	50.86	49.85	45.40
128	16	8×	32×	48.17	37.31	53.93	51.33	47.69
128	32	4×	16×	48.77	37.96	54.71	52.07	48.38
128	64	2×	8×	48.17	37.57	53.22	51.78	47.68
256	8	32×	128×	44.65	34.73	50.31	46.67	44.09
256	16	16×	64×	46.15	38.22	53.21	49.11	46.67
256	32	8×	32×	48.99	38.86	54.36	51.85	48.52
256	64	4×	16×	48.84	39.96	54.59	51.52	48.73
512	64	8×	32×	48.32	39.44	55.11	51.48	48.59
1024	128	8×	32×	48.47	38.73	54.77	52.11	48.52

Table 5: StateVid observation and readout budget sweep. T is the observed-frame cap and K is the number of ordered snapshots. Obs./readout is T/K and Token ratio is $4T/K$ after shared 2×2 spatial downsampling. Scores are percentages; Avg. is the unweighted mean of the four benchmarks.

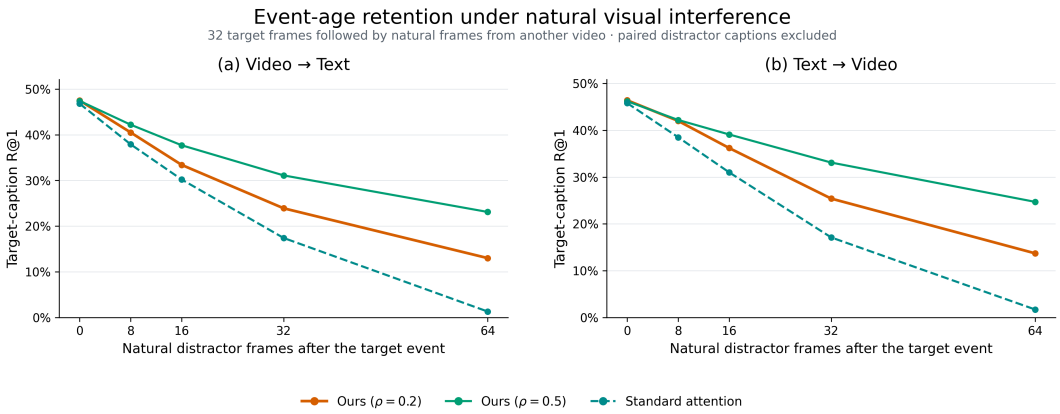

Figure 10: Target-pair retrieval under natural visual interference. A 32-frame target video is followed by up to 64 frames from another video. The final representation is evaluated for StateVid with $\rho = 0.2$ or 0.5 and for standard attention; paired distractor captions are excluded from the candidate set.

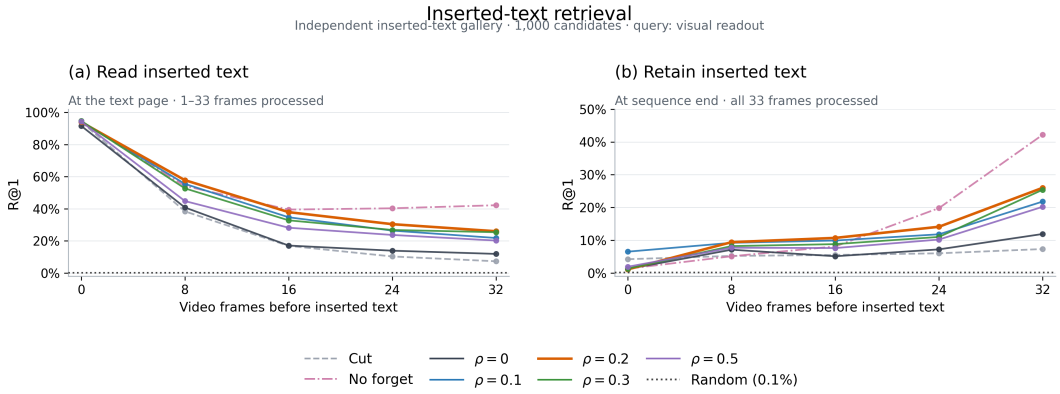

Figure 11: Reading and retaining a rendered text page inserted at different positions in a 33-frame sequence. Retrieval uses an independent gallery of 1,000 inserted texts. The left panel reads at the inserted page; the right panel reads at the sequence end. Random performance is 0.1%. The legend uses α for historical-target decay ρ.

T	K	Obs./readout	Token ratio	LongVideoBench	LV-Bench	MLVU	VideoMME	Avg.
32	8	4×	16×	43.98	37.06	49.21	47.07	44.33
32	16	2×	8×	45.92	36.41	51.69	50.00	46.01
32	32	1×	4×	47.27	37.96	52.62	50.85	47.18
64	8	8×	32×	44.58	35.38	49.32	47.56	44.21
64	16	4×	16×	45.25	37.51	50.62	50.30	45.92
64	32	2×	8×	46.52	37.64	51.10	50.93	46.55
64	64	1×	4×	46.45	39.96	52.65	52.00	47.76
128	8	16×	64×	43.68	34.86	49.59	46.89	43.75
128	16	8×	32×	44.28	37.70	50.63	48.78	45.35
128	32	4×	16×	45.48	38.28	51.57	50.44	46.44
128	64	2×	8×	46.30	39.25	52.12	50.93	47.15
256	8	32×	128×	43.90	34.47	49.88	45.93	43.55
256	16	16×	64×	43.90	35.89	51.00	48.37	44.79
256	32	8×	32×	44.58	36.41	51.33	50.33	45.66
256	64	4×	16×	44.88	38.48	52.87	50.78	46.75

Table 6: Mean-pooling observation and readout budget sweep. Obs./readout and Token ratio follow the definitions in Table 5. Scores use the same benchmarks and aggregation.

T	K	LongVideoBench	LV-Bench	MLVU	VideoMME	Avg.
8	8	46.30	34.28	49.21	47.04	44.21
16	16	47.42	37.70	50.43	50.67	46.55
32	32	48.84	37.19	51.70	51.48	47.30
64	64	50.11	39.83	52.49	52.37	48.70

Table 7: Complete frame-wise SigLIP baseline budget sweep. Here $T = K$.

E Delta-Rule Updates versus Softmax Aggregation

Repeated video observations expose how an update responds to a later change. We analyze one spatial position and attention head with fixed scores, keys, and gates. The comparison isolates the response of softmax aggregation and gated delta updates. We then trace retention across later updates and training clip boundaries.

Repeated observations compete for softmax weight. For a fixed query q, let $s_j = q^\top k_j/\sqrt{d}$ be the attention score. Consider M repeated observations with value v_0 and score s_0, followed by one changed observation with value $v_\star = v_0 + \delta$ and score $s_\star$. The softmax readout is

$$o_{\mathrm{attn}} = v_0 + w_\star \delta, \qquad w_\star = \frac{\exp(s_\star)}{M\exp(s_0) + \exp(s_\star)} = \frac{1}{1 + M\exp(-\Delta)}, \qquad \Delta = s_\star - s_0. \tag{14}$$

For a fixed score gap Δ, the coefficient of the change decays as $O(1/M)$. Keeping $w_\star \geq \eta$, for a fixed $\eta \in (0,1)$, requires $\Delta \geq \log M + \log[\eta/(1-\eta)]$. Under fixed scores, the changed observation needs a score advantage that grows with $\log M$ to retain the same weight. Learned scores and temporal biases may change this behavior.

Repeated delta-rule updates approach a bounded association. Rearranging Eq. 2 gives

$$S_t = \alpha_t S_{t-1} + \beta_t\big(v_t - \alpha_t S_{t-1} k_t\big)k_t^\top. \tag{15}$$

The write depends on the prediction residual after decay. To examine repetitions, assume a shared unit-norm key k, a constant value v_0, and constant gates $\alpha, \beta \in (0,1)$. Writing $z_t = S_t k$ and $a = \alpha(1-\beta)$ yields

$$\begin{aligned} z_t &= a z_{t-1} + \beta v_0, \\ z_M &= a^M z_0 + \frac{\beta(1-a^M)}{1-a} v_0 \;\longrightarrow\; z_\infty = \frac{\beta}{1-a} v_0. \end{aligned} \tag{16}$$

Since $0 < a < 1$, the association approaches a bounded fixed point, and its net change decreases geometrically with repetition. For $\alpha < 1$, writing can remain active at the fixed point to compensate for decay. At $\alpha = 1$, $z_\infty = v_0$ and the residual vanishes. With the normalized key used here, $(I - \beta k k^\top)$ attenuates the existing association along k and is generally a nonorthogonal operator.

Response to a new change. After the repeated prefix, compare two possible next observations with values v_0 and $v_0 + \delta$. Hold their key $k_\star$, gates $\alpha_\star, \beta_\star$, and readout query q fixed. The two updates share the same prefix state S_M, so their readout difference is exactly

$$\Delta o_{\mathrm{delta}} = \big(S_{M+1}^{\mathrm{changed}} - S_{M+1}^{\mathrm{unchanged}}\big)q = \beta_\star \delta\,(k_\star^\top q). \tag{17}$$

For fixed gates and key-query alignment, this response is independent of M. In the shared-key case $k_\star = q = k$ with $\|k\|_2 = 1$, standard content-only softmax attention assigns the changed observation weight $1/(M+1)$, while Eq. 17 gives coefficient $\beta_\star$. The delta readout therefore has a larger response to the same value change whenever $M > 1/\beta_\star - 1$. This fixed-key example gives the delta update a stronger response after sufficiently many repeated observations.

Together, Eqs. 16 and 17 show a bounded association under repetition and a gate-controlled response to a subsequent change. In the learned encoder, keys, gates, and queries vary with the input, while later updates and finite state capacity affect retention. The experiments above measure the resulting behavior.

Retention along the update path. For general keys and gates, the scalar decay accumulated between steps j and i is

$$F_{ij} = \prod_{t=j+1}^{i} \alpha_t = \exp\left(\sum_{t=j+1}^{i} g_t\right), \qquad \alpha_t = \exp(g_t). \tag{18}$$

Conditioned on the keys and gates, the contribution of the write at j to S_i is $\beta_j F_{ij} v_j k_j^\top P_{ij}$, where $P_{ij} = \prod_{t=j+1}^{i}(I - \beta_t k_t k_t^\top)$ and the matrices are ordered from left to right with increasing t. Retention therefore includes scalar decay and revisions along later keys. A softmax logit bias $b_{ij} = \log F_{ij}$ represents the scalar factor. The recurrent update also applies the key-dependent transitions P_{ij} as observations arrive.

Boundary-enhanced decay. During the training described in Sec. 4.2, we rescale the log-decay at a clip boundary as $g_r \leftarrow \kappa g_r$ with $\kappa > 1$. Holding the predicted gate sequence fixed to isolate this intervention, the ratio of the modified to the original scalar retention factor is

$$\frac{F'_{ij}}{F_{ij}} = \exp\left((\kappa - 1) \sum_{r \in \mathcal{B}(i,j)} g_r\right) = \prod_{r \in \mathcal{B}(i,j)} \alpha_r^{\kappa - 1}, \tag{19}$$

where $\mathcal{B}(i, j)$ contains the clip boundaries satisfying $j < r \leq i$. At each crossed boundary, the new retention gate is α_r^κ; the additional attenuation relative to the original gate is $\alpha_r^{\kappa-1}$. For $\kappa = 3$, these are α_r^3 and α_r^2, respectively. Because $0 < \alpha_r < 1$, the intervention weakens carried history while preserving a continuous transition. Its strength depends on the predicted gate at the boundary. At inference, the model uses its learned gates without this boundary intervention.

F Efficiency Analysis

StateVid processes each observed frame and exposes selected state snapshots to the language model. We measure the resulting encoder and prefill costs separately. The forward-pass FLOP estimates exclude video decoding, data transfer, and autoregressive generation.

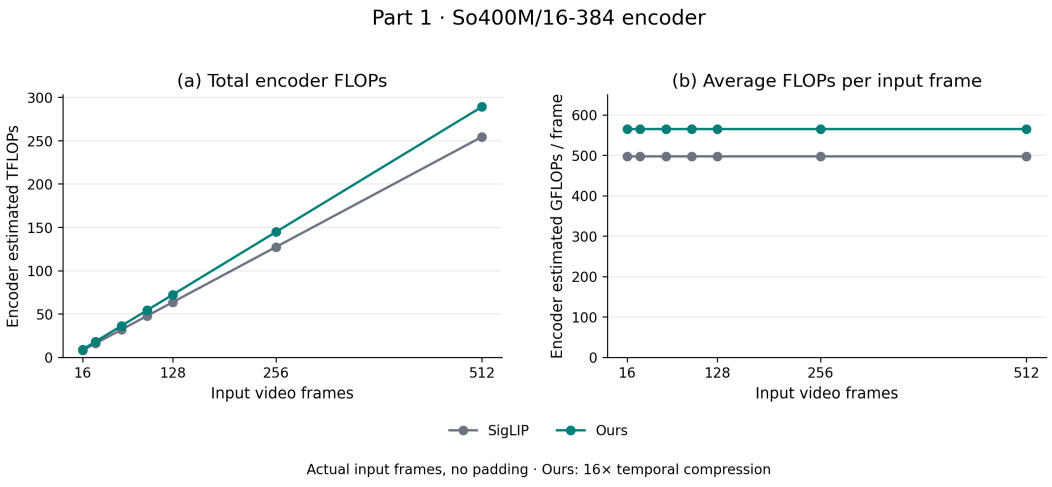

Figure 12: Estimated FLOPs of the So400M/16-384 visual encoder as the number of observed frames increases. Values use the actual frame count without padding; StateVid's per-frame overhead remains approximately constant.

Encoder cost. Figure 12 compares StateVid with frame-wise SigLIP. Both costs grow linearly with observed frames. StateVid requires approximately 566 GFLOPs per frame versus 496 GFLOPs for SigLIP, a 14% overhead for temporal state updates. At 512 frames, the estimates are 290 and 254 TFLOPs.

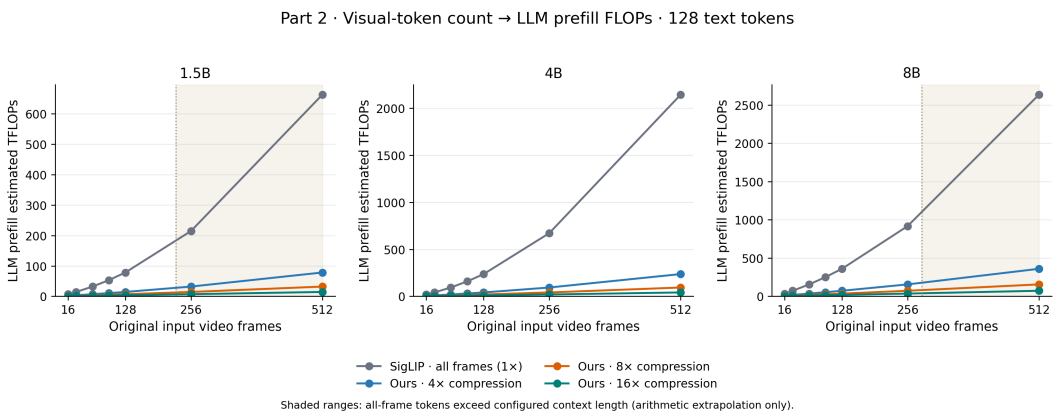

Figure 13: Estimated language-model prefill FLOPs as a function of the original video length for 1.5B, 4B, and 8B language models. All settings include 128 text tokens. Sparse state readout uses $4\times$, $8\times$, or $16\times$ fewer visual tokens. Shaded regions indicate where the all-frame visual sequence exceeds the configured context length; values there are arithmetic extrapolations rather than executable runs.

Language-model prefill. Figure 13 shows lower prefill FLOPs as fewer state snapshots are exposed, across all three language-model sizes. The gap grows with video length as the all-frame visual sequence expands. StateVid therefore trades higher visual-encoder cost for a shorter LLM visual context; the figure characterizes prefill cost rather than end-to-end latency.

G Additional Details

This section records the data construction, training settings, and evaluation protocols for the encoder and VideoLLM. Encoder comparisons share preprocessing, sampled frames, prompts, and evaluation code. Encoder results use one checkpoint and zero-shot evaluation.

For Table 1, we rerun the SigLIP2 baselines and the PE_{core} results on DiDeMo. All other external baseline entries are reproduced from the cited papers and are included for context rather than as protocol-matched reruns.

G.1 Construction of Recaptioned Video Data

Source data and annotation format. The encoder training set contains 2M videos sampled from InternVid-10M, as specified in Sec. 5. We use Qwen3-VL-4B to generate descriptions of their visual content. The annotation scripts read Parquet records containing video bytes and retain the existing metadata alongside the generated text and its prompt. One annotation mode produces a motion-adaptive description, and the other produces two short retrieval captions from each source record.

Prompt lists. For reproducibility, Tab. 8 lists the prompt pools used by the scripts verbatim. The dynamic pool is used for motion-adaptive videos and is also the pool from which the two short retrieval-caption prompts are sampled. The static pool is used when the motion score is below the threshold.

Motion-adaptive descriptions. The motion-adaptive mode samples every tenth decoded frame, resizes each sampled frame to 64×64 pixels, and converts it to grayscale. Its motion score is the mean absolute pixel difference between consecutive sampled frames, averaged over all available pairs on the 0–255 intensity scale. A score below 2.5 selects a prompt from five static-scene templates; otherwise, the script selects from eleven action-oriented templates. For example, the static pool includes "Describe the visual content of this static scene," while the action pool includes "What is the main subject doing? Answer in a single concise sentence." The selected prompt is prefixed with "Watch the video and reply in brief:". The output record stores the generated description, prompt, motion score, and static/dynamic tag. The motion score selects a prompt category.

Short captions for retrieval. The retrieval-caption mode independently samples two prompts from the eleven short-caption templates. These templates emphasize the main subject, visible action, and concise subject–verb–object descriptions; each request also appends "Reply to me with the briefest description." For each caption, the script allows up to five attempts, resampling the prompt on each attempt, and accepts a nonempty response containing at most 60 whitespace-delimited words. Both caption–prompt pairs are stored with the video, and rows missing either accepted caption are removed. The programmatic length threshold is 60 words. This mode accepts both static and dynamic videos.

Caption generation settings. Both modes use the same video-language inference interface. For a video of duration d seconds, the requested sampling rate is $f = \max(0.1, \min(128/d, 1.0))$ fps; the fallback rate is 1 fps if duration probing fails. This targets approximately 128 frames for longer clips. The 0.1 fps lower bound can yield more frames. Generation uses temperature 0.7, top-p 0.95, top-k 20, presence penalty 1.5, and a maximum of 100 output tokens, with thinking disabled. Each API request allows two retries after the initial attempt. A length-truncated response triggers a retry with an additional brevity instruction. These annotation-time sampling settings are distinct from the 2 fps decoding and 8–24-frame sampling used for encoder training.

Storage and inspection. The annotated Parquet files are converted into WebDataset shards, with each sample containing an MP4 video and a JSON record of all non-video fields. Sample keys combine the source Parquet filename and row index, and the source filename is also retained in the metadata. The command-line defaults limit each shard to 10,000 samples or approximately 3 GiB. The video compression utility supports a 2 fps training copy at a height of 360 pixels, preserving aspect ratio and encoding with H.264 at CRF 28 using the `veryfast` preset. Other archive entries are copied unchanged. A Parquet viewer displays videos beside their annotation fields for qualitative inspection.

Table 8: Prompt pools used for video recaptioning.

Static-scene pool (5 prompts)

- `Describe the visual content of this static scene.`
- `Describe the scene as a still visual presentation.`
- `Describe the overall appearance of the video frame.`
- `Provide a factual description of the scene's content.`
- `What are the key visual components of this video?`

Dynamic/short-caption pool (11 prompts)

- `Summarize the visual content in a single, short sentence.`
- `Describe the main action in one sentence under 20 words.`
- `What is the main subject doing? Answer in a single concise sentence.`
- `Provide a one-sentence factual caption of the scene.`
- `Use the format 'A [subject] is [action] [object/location]' to describe the video.`
- `Write a concise caption: Subject + Verb + Object (e.g., A man is driving a car).`
- `Create a brief, keyword-focused caption for a video search engine.`
- `Describe the video action concisely, ignoring background details.`
- `Summarize the visible movement in less than 15 words.`
- `Write a short description in the present continuous tense (is/are doing).`
- `Briefly state what happens in the video using simple language.`

G.2 Vision Encoder Training

Architecture and initialization. We initialize the spatial backbone from either SigLIP2-B/16-224 or SigLIP2-So400M/16-384 and retain its corresponding input resolution. Four temporal blocks are inserted after the last four spatial blocks. Both the Temporal Layer and State Layer use causal GDR computation with an MLP expansion ratio of 2, a state expansion factor of 1, and a causal convolution kernel size of 4. The SigLIP2 text encoder is frozen throughout encoder training and is used to encode the retrieval targets. The temporal modules are trained together with the unfrozen visual parameters, whereas the text parameters are never updated.

Temporal aggregation baselines. The *Attn* entries in Tab. 3 and Fig. 3 use learned attention pooling at each spatial position. The attention-plus-mean variant applies causal temporal attention before averaging frame features. GDR and attention share the backbone width and head configuration, temporal block locations, input features, MLP expansion ratio, training data, sampled frames, update count, and optimizer schedule. The mechanism analysis in Sec. E holds scores fixed. The empirical Attn baseline learns its projections jointly with the visual encoder.

Data construction. The image portion combines CC12M, ImageNet-21K, and recaptioned versions of these data. The video portion contains 2M videos sampled from InternVid-10M; we recaption these videos with Qwen3-VL-4B using the procedures described in Sec. G.1. Videos are decoded at 2 fps, with between 8 and 24 frames retained per clip. A video batch contains 20 clips, which are concatenated in their sampled order into a stream of at most 480 frames as described in Sec. 4.2. The recurrent state is not reset between clips, but the clip endpoints are recorded for retrieval supervision. Image and video updates are interleaved using one image update for every four video updates. Image and video micro-batch sizes are 64 and 10, respectively.

Objectives. For image batches, we use the three distillation terms in Eq. 10. We set $(\lambda_{\cos}, \lambda_{\text{rel}}, \lambda_{\text{img-ret}}) = (1, 0.01, 1)$ and the overall image-distillation loss weight to 1. The terms align token features, spatial relations, and image–text retrieval distributions to preserve the pre-

trained image representation. For video batches, normalized endpoint states and text features are compared with a sigmoid retrieval loss. The similarity logit is

$$\ell_{ij} = 100\,\mathbf{v}_i^\top \mathbf{t}_j - 15. \tag{20}$$

For clips within the same concatenated stream, the target for endpoint i and caption j is

$$y_{ij} = \begin{cases} 1, & i = j, \\ \rho^{i-j}, & j < i, \\ 0, & j > i, \end{cases} \qquad \rho = 0.2, \tag{21}$$

where temporal distance counts clips. Texts from other streams and a queue of 8,192 text features are negatives. We use $\kappa = 3$ for boundary-enhanced decay during training. At inference, the learned gates operate without boundary intervention. Figure 5 reports the sweep $\rho \in \{0, 0.1, 0.2, 0.3, 0.5\}$ alongside hard-reset and no-boundary-enhancement controls.

Optimization. We train the encoder for 20K optimizer steps using AdamW, a peak learning rate of 5×10^{-4}, weight decay 0.05, cosine decay, and a 3% linear warmup. Training uses BF16, gradient checkpointing, gradient clipping at 1.0, and a fixed random seed of 42. All ablations use the same initialization, data, update count, and optimization schedule unless the ablated component itself changes one of these quantities.

G.3 Zero-Shot Encoder Evaluation

Video–text retrieval. We evaluate on MSR-VTT (Xu et al., 2016), DiDeMo (Anne Hendricks et al., 2017), and ActivityNet Captions (Krishna et al., 2017). For the main table, we uniformly sample 24 frames over each video. For the frame-count study in Fig. 3, we use $T \in \{16, 24, 32, 40, 48, 56, 64, 80, 96\}$ over the same video duration. All methods receive the same decoded frames for each setting. StateVid uses the endpoint state at the last valid frame, and baseline methods aggregate their corresponding frame features. We ℓ_2-normalize video and text features and rank the complete test candidate set by cosine similarity. The main paper reports Recall@1 for text-to-video and video-to-text retrieval, denoted T2V and V2T.

Action recognition. Kinetics-400 (Kay et al., 2017), UCF101 (Soomro et al., 2012), and HMDB51 (Kuehne et al., 2011) are evaluated as zero-shot classification tasks. Class names are expanded with the same action prompt templates for every model and encoded by the frozen SigLIP2 text encoder. Template features are averaged per class and normalized. Predictions are obtained from cosine similarities between the video representation and these class prototypes; we report top-1 accuracy.

Image classification. The 11 datasets in Tab. 2 are ImageNet-1K, CIFAR-10, CIFAR-100, Caltech-101, PASCAL VOC 2007 Classification, Stanford Cars, Oxford-IIIT Pets, Food-101, Oxford Flowers-102, EuroSAT, and RESISC45. Dataset-specific prompt templates are encoded once by the frozen text encoder, averaged within each class, and normalized. Each image passes through the video interface as a static input. We classify its normalized representation by exact cosine similarity to all class prototypes and report top-1 accuracy over the full evaluation split.

G.4 VideoLLM Training

We use Qwen2.5-1.5B as the language-model backbone. A learned projector connects it to the pretrained StateVid encoder, and we train the resulting VideoLLM in three stages. Stage 1 warms up only the projector on LLaVA-ReCap-558K (Liu et al., 2024a); the vision encoder and language model remain frozen. The projector learning rate is 1×10^{-3}, and training lasts for one epoch. Stage 2 performs short-video alignment on approximately 0.3M videos mixed with 0.2M images, using at most 32 frames per video. Stage 3 performs long-video instruction tuning on approximately 0.4M videos mixed with 0.2M images, including FineVideo (Farré et al., 2024) and LLaVA-Video data. It observes up to 256 frames and uses sparse state snapshots to keep the visual token budget independent of the number of observed frames.

In Stages 2–3, the language model, projector, and temporal layers are jointly optimized, while the spatial vision backbone remains frozen. Both stages use a learning rate of 2×10^{-5}, a global batch size of 64, and one epoch of training. Across all stages we use AdamW, BF16, gradient

checkpointing, a cosine learning-rate schedule, and a 1% linear warmup. Image examples use a single projected image representation. For videos, the state snapshots are selected in chronological order using Eq. 7 and projected into the language-model embedding space before being interleaved with the text tokens.

G.5 VIDEOLLM EVALUATION

We evaluate with a local version of `lmms-eval` on LongVideoBench (Wu et al., 2024), LVBench (Wang et al., 2025a), MLVU (Zhou et al., 2025), PerceptionTest (Patraucean et al., 2023), and VideoMME (Fu et al., 2025). The main protocol samples at 2 fps with $T \leq 256$, while the budget sweep extends T to 1,024. We use T for the observation cap and K for ordered LLM readouts, each with the same spatial grid. Tab. 4 fixes $K{=}32$; frame-wise SigLIP2 selects $T{=}K$ frames, Mean and similarity-based mean aggregate up to 256 observations, ToMe merges similar visual tokens to the same readout budget, and StateVid selects K snapshots from its evolving state. For Fig. 4, Long avg. is the equal-weight mean of LongVideoBench, LVBench, MLVU, and VideoMME-long, excluding PerceptionTest. Panel a fixes $K{=}32$ and varies T, while Panel b uses $T = 4K$ for StateVid and Mean and $T = K$ for frame-wise SigLIP2.

For multiple-choice benchmarks, each method receives the same question, answer candidates, benchmark-specific prompt, and generation settings. The common evaluation code parses the generated option and applies official benchmark scoring. Evaluation uses a per-process batch size of 2 and eight parallel processes. Predictions are merged by sample identifier. We archive frames at 4 fps with at most 1,024 frames per video, a maximum long side of 384 pixels, and JPEG quality 85. Model inputs follow the shared 2 fps and maximum-frame rules when sampled from this archive.