WIRE: WEIGHT SPACE IMPLICIT REPRESENTATION ENCODING OF FROZEN VIDEO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video language models represent a video as a long sequence of visual embeddings in a decoder's input. Every visual embedding is processed by every decoder layer during prefill, stored in the key value cache, and attended by every generated token, so a long video raises prefill, memory, and generation cost at once. We present WIRE (*Weight Space Implicit Representation Encoding*), which reduces this recurring cost by moving part of the video conditioned computation of a frozen video language model from its input activations into a temporary low rank update of its weights. WIRE keeps a short explicit visual context, so the update acts as a correction rather than a replacement of the video. WIRE first compresses the full resolution embeddings of each frame into a small set of *carrier tokens* using a question conditioned bidirectional selective scan, and physically contracts the decoder input so every decoder layer, cache entry, and decoding step operates on the shorter sequence. Hypernetworks at selected decoder layers read the compressed forward pass together with a readout of the full resolution features removed by contraction and generate LoRA updates to the attention and feedforward projections. The updates are merged after prefill, so autoregressive decoding runs the unmodified frozen backbone. On Qwen3-VL-8B, WIRE with 16 carrier tokens per frame contracts the visual stream by $3.5\times$, reduces decoder prefill time by $3.35\times$ at 256 frames and $4.14\times$ at 1024 frames, shrinks the key value cache proportionally, and keeps per token decoding latency approximately constant. Across nine benchmark settings at 256 frames, it obtains the highest average accuracy among efficient video language models we evaluate under a common 256 frame budget. A paired update analysis toggles only the generated update while holding the compressed visual context fixed. Applying the generated update raises whole benchmark accuracy on Video-MME, MVBench, and LongVideoBench by up to $+4.4$ points, with the largest gains on compression sensitive capabilities such as counting and action count, establishing that the generated update does task relevant work beyond the carrier tokens.

1 INTRODUCTION

Video language models represent a sampled video as visual embeddings in the input sequence of a decoder only language model (Zhang et al., 2025d; Bai et al., 2025a; Li et al., 2025; 2026; Cheng et al., 2024; Ye et al., 2025a; Liu et al., 2024a; Zhang et al., 2025a): a vision encoder maps each frame to a spatial grid of embeddings, interleaved with timestamps and question text. This simple, effective interface has a structural cost. Every visual embedding is processed by every decoder layer during prefill, occupies a key value (KV) cache position, and is attended by every output token, so a video that occupies thousands of input positions inflates prefill compute, KV cache memory, and per token generation cost all at once.

Existing efficient video language models cut this cost by pruning, merging, resampling, or sparsifying the visual sequence, or by selecting frames (Chen et al., 2024; Yang et al., 2024; Islam et al., 2025; Shen et al., 2025; Tao et al., 2025; Ren et al., 2025; Song et al., 2026; Qin et al., 2025; Jiang et al., 2025). These shorten the decoder input, but all retained visual information still flows through the activation stream, is reprocessed by the full decoder, and must contain enough to answer every question. At strong compression, a compact visual sequence can lose information needed for

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

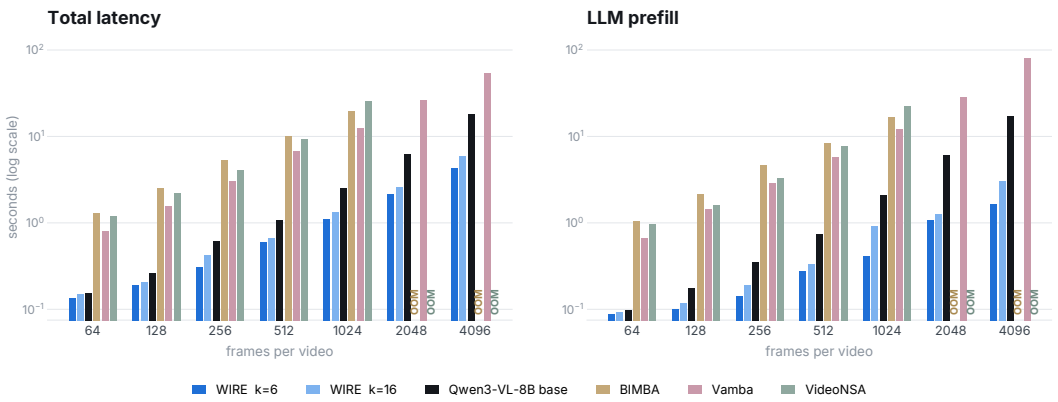


Figure 1: **Latency versus video length.** Times are measured on one B200 GPU at batch size one in bfloat16, excluding video file decoding. *Left:* total model time to first token. *Right:* decoder prefill only. WIRE contracts the sequence before the first decoder layer, so its advantage grows with frame count and it continues to run beyond input lengths where several baselines exhaust memory.

counting, action recognition, temporal ordering, and questions grounded in a particular subtitle or event.

WIRE (*Weight Space Implicit Representation Encoding*) explores a complementary pathway, a temporary input conditioned weight update that carries a video conditioned correction rather than a standalone representation of the video. Unlike activations, whose cost grows with sequence length and output tokens, such an update is computed once and merged before decoding.

Let V denote a video, q its question, X the original multimodal decoder input, $\mathcal{C}$ a visual context compressor, and f_θ a frozen video language model. WIRE learns a compressor and a video question conditioned update $\Delta\theta(V, q)$ such that

$$f_{\theta+\Delta\theta(V,q)}\big(y \mid \mathcal{C}(X)\big) \approx f_\theta\big(y \mid X\big). \tag{1}$$

The contracted input $\mathcal{C}(X)$ retains a small, spatially organized set of visual *carrier tokens*, and the generated update changes how the frozen decoder processes them and the question.

Our contributions are threefold. First, we introduce a frozen video LLM architecture that pairs physical visual token contraction with per example weight space adaptation, so sequence length reduction reaches every decoder layer, cache entry, and decoding step, separating WIRE both from methods that keep all retained evidence in the activation stream. Second, we train the compressor and hypernetworks by context distillation (Hinton et al., 2015; Snell et al., 2022) from the same frozen backbone on the uncompressed input. Third, we provide an accuracy versus latency frontier across nine video benchmark settings and a large paired update analysis (Section 5.2) that toggles only the generated update at fixed compressed context, measuring whether it does task relevant work beyond the carrier tokens; whether that work is video specific we leave as a hypothesis. Figure 1 shows that because WIRE contracts the sequence before the first decoder layer, its prefill and time to first token advantage widens with frame count.

2 Related Work

Efficient video language models. Most efficient video language models shorten the decoder sequence by selecting, merging, or pooling visual embeddings (Chen et al., 2024; Yang et al., 2024; Huang et al., 2025; Tao et al., 2025; Liu et al., 2026a; Bolya et al., 2023; Zhang et al., 2025c; Xing et al., 2025; Shao et al., 2026), by resampling into a few carrier tokens (Jaegle et al., 2021; Alayrac et al., 2022; Li et al., 2023; Mu et al., 2023; Petrov et al., 2025), or by adapting spatiotemporal token allocation under a budget (Ye et al., 2025b; Shen et al., 2025; Cai et al., 2025; Zhang et al., 2025e; Zohar et al., 2025; Liu et al., 2024b; Zhang et al., 2025b; Li et al., 2025). A second family keeps the visual sequence but cuts its attention or key value cache cost, sparsifying visual attention or compressing the cache mid decoder (Song et al., 2026; Tu et al., 2025; Lin et al., 2025; Wang et al., 2025b; Qin et al., 2025; Xiao et al., 2023; Zhang et al., 2023; Kim et al., 2025). A third,

complementary line reduces *which* frames reach the decoder, via temporal projectors that propagate information before pooling (Jiang et al., 2025), training free frame selection (Kim et al., 2026; Tang et al., 2025), or agentic coarse to fine zooming for high resolution clips on demand (Ding et al., 2026). All leave the retained evidence in the activation stream, reprocessed by the full decoder for every question. WIRE instead removes visual positions before the first decoder layer, adding a sample specific weight space pathway that recovers the lost evidence.

Selective state space models for video. Selective state space models give linear time sequence mixing (Gu et al., 2022; Gu & Dao, 2024; Dao & Gu, 2024; Lahoti et al., 2026) and appear in video encoders, hybrid decoders, and temporal enrichment modules (Li et al., 2024a; Ren et al., 2025; Jiang et al., 2025; Liao et al., 2025). BIMBA interleaves learned carrier vectors with visual embeddings and refines them with a selective scan before decoding (Islam et al., 2025). Our compressor shares this mixing principle but adds question embeddings at both scan boundaries, preserves retained carriers' original multimodal rotary positions, and passes the full resolution stream to a hypernetwork after contraction. We do not claim it as a primary contribution: it is deliberately BIMBA like, so our novelty lies in the contraction and weight correction mechanism and in the paired analysis of what the generated update adds.

Generated adapters and contextual weights. Hypernetworks generate another network's parameters (Ha et al., 2017), and LoRA makes input conditioned updates tractable at scale (Hu et al., 2022). Recent work generates adapters from task descriptions, documents, edit instructions, and prefixes (Li & Liang, 2021; Karimi Mahabadi et al., 2021; Charakorn et al., 2025; 2026; Zeng et al., 2026a; Tan et al., 2026; Abdalla et al., 2026; Zeng et al., 2026b; Liu et al., 2026b), while context distillation and fast weight methods store a context outside the activation stream (Askell et al., 2021; Snell et al., 2022; Eyuboglu et al., 2026; Wang et al., 2026; Shin et al., 2025; Behrouz et al., 2026; Zhang et al., 2026; Ba et al., 2016; Schmidhuber, 1992). All of these methods condition on text. WIRE instead generates its adapter from a compressed video forward pass and binds generation to physical sequence contraction. Mechanically its update is a video conditioned, hypernetwork generated LoRA in this contextual adapter family (Han et al., 2026; Zeng et al., 2025). To our knowledge it is the first to couple video conditioned adapter generation with physical contraction. The adapter is produced in one amortized forward pass and compensates the frozen decoder for the removed positions.

3 METHOD

WIRE compresses the visual prefix that enters a frozen video language model and reuses the removed information to generate a temporary video conditioned weight update.

3.1 PROBLEM SETUP AND NOTATION

Let a video be a sequence of T sampled frames, $V = (I_1, \ldots, I_T)$, and let q denote the text question. A frozen vision encoder followed by the frozen multimodal projector maps frame I_t to a matrix of visual tokens,

$$\mathbf{X}_t \in \mathbb{R}^{n_t \times d}, \qquad t \in \{1, \ldots, T\}, \tag{2}$$

where d is the language model hidden width and n_t is the number of full resolution visual tokens for frame t. Bold uppercase letters denote matrices and bold lowercase letters denote vectors. Let n_{ref} be the nominal number of visual tokens per frame under the chosen visual preprocessing, and $n_{\text{vis}} = \sum_{t=1}^{T} n_t$ the total number of full resolution visual tokens. Backbone and preprocessing are specified in the experimental protocol.

Let $\mathbf{Q} \in \mathbb{R}^{n_q \times d}$ contain the projected timestamp and question embeddings. The original multimodal prefix is the temporally ordered concatenation of the frame token matrices and the text embeddings, $\mathbf{X} \in \mathbb{R}^{S \times d}$ with $S = n_{\text{vis}} + n_q$. The frozen decoder is f_θ, with parameters θ. Neither the vision encoder nor f_θ is updated.

The method builds a shorter prefix $\widetilde{\mathbf{X}}$ and, for selected decoder layers, a video question conditioned update $\Delta\theta(V, q)$. Its training goal is for the frozen decoder with the short prefix and temporary

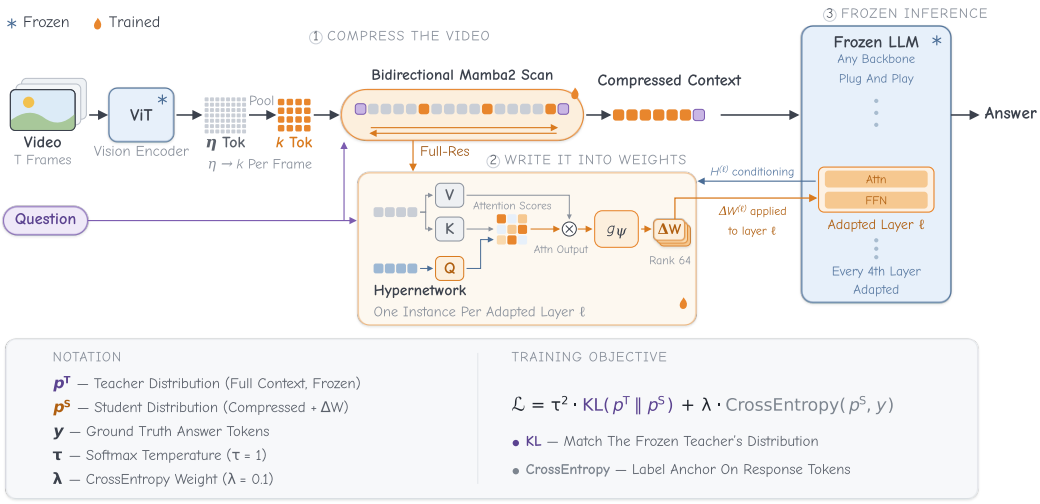


Figure 2: **WIRE overview.** The frozen vision encoder produces n_t full resolution visual tokens for frame t. Bilinear resampling initializes k_t carrier tokens, and a bidirectional Mamba2 scan refines the carrier tokens while reading the full resolution visual stream and the question. The carrier tokens replace the original frame tokens in the decoder prefix. At selected decoder layers, a hypernetwork uses the contracted decoder state and a full resolution per frame readout to generate LoRA updates for selected attention and MLP projections. The updates are merged after prefill. Decoding then uses the frozen backbone over the contracted prefix.

update to reproduce the behavior of the same decoder on the original prefix:

$$f_{\theta+\Delta\theta(V,q)}\left(y \mid \widetilde{\mathbf{X}}\right) \approx f_\theta(y \mid \mathbf{X})\,. \tag{3}$$

This is the setup-level counterpart of the objective in Eq. (1): f_θ realizes the conditional π_θ, and $\widetilde{\mathbf{X}}$ is the counterpart of the compressed input $\mathcal{C}(X)$. We henceforth write f_θ for the frozen video language model.

We distinguish two visual representations. A *full resolution visual token* is a row of $\mathbf{X}_t$. A *carrier token* is a learned replacement for a group of full resolution visual tokens that remains in the decoder input after contraction.

Figure 2 gives an overview.

3.2 QUESTION CONDITIONED VISUAL COMPRESSION

Carrier token initialization. We assign k_t carrier tokens to frame t. Let k be the target carrier token budget for a frame with n_{ref} full resolution tokens, scaled to the actual frame length as

$$k_t = \min\left\{n_t,\ \max\left\{1,\ \operatorname{round}\left(\frac{n_t}{n_{\text{ref}}}k\right)\right\}\right\}. \tag{4}$$

The token budget and resulting compression ratio are experimental configuration choice, reported in Section 4.

To preserve spatial layout at initialization, we reshape the n_t tokens of a frame to their original $h_t \times w_t$ grid ($n_t = h_t w_t$) and choose integers a_t, b_t with $a_t b_t = k_t$ and a_t/b_t closest to h_t/w_t. Bilinear resampling then initializes the carrier token matrix,

$$\mathbf{C}_t^{(0)} = \operatorname{Bilinear}\big(\operatorname{grid}_{h_t \times w_t}(\mathbf{X}_t),\ a_t, b_t\big) \in \mathbb{R}^{k_t \times d}, \tag{5}$$

where $\operatorname{grid}_{h_t \times w_t}$ arranges the $n_t = h_t w_t$ tokens of $\mathbf{X}_t$ onto their spatial grid. Bilinear is standard two dimensional bilinear interpolation applied independently to each of the d channels: it treats the $h_t \times w_t$ token grid as a continuous field and resamples it onto a coarser $a_t \times b_t$ grid, so each carrier token is a convex combination of the four full resolution tokens nearest its sampling location, with weights given by that location's fractional distance to those tokens. Flattening the $a_t \times b_t$ output

back to a sequence gives the $k_t = a_t b_t$ carrier rows. This initialization is parameter free. Learned compression comes only from the selective scan below.

Bidirectional selective scan. The initial carrier tokens summarize local spatial neighborhoods but have not seen the rest of the video or the question, so we process the full resolution tokens, the carrier tokens, and the question jointly. Let $[\,\cdot\,;\,\cdot\,]$ denote sequence concatenation and $\mathrm{Interleave}(\mathbf{X}_t, \mathbf{C}_t^{(0)})$ insert the k_t carrier tokens approximately uniformly among the n_t full resolution tokens of frame t. The scan input is

$$\mathbf{Z} = \Big[\mathbf{Q};\ \mathrm{Interleave}(\mathbf{X}_1, \mathbf{C}_1^{(0)});\ \cdots\ ;\ \mathrm{Interleave}(\mathbf{X}_T, \mathbf{C}_T^{(0)});\ \mathbf{Q}\Big]\,. \tag{6}$$

Placing $\mathbf{Q}$ at both ends ensures that, in at least one scan direction, every carrier token is close to the question tokens.

A forward and a reverse Mamba2 block, with independent parameters, produce

$$\mathbf{Y} = \mathbf{Z} + \mathrm{SSM}_{\rightarrow}(\mathrm{LN}(\mathbf{Z})) + \mathrm{Reverse}[\mathrm{SSM}_{\leftarrow}(\mathrm{LN}(\mathrm{Reverse}[\mathbf{Z}]))]\,. \tag{7}$$

The carrier token rows of $\mathbf{Y}$ for frame t form $\mathbf{C}_t \in \mathbb{R}^{k_t \times d}$. The remaining visual token rows form $\mathbf{F}_t \in \mathbb{R}^{n_t \times d}$. The carrier tokens enter the contracted decoder prefix. The full resolution features $\mathbf{F}_t$ stay outside it, used only by the hypernetwork, so contraction removes visual positions while the hypernetwork can still consult the full resolution evidence.

3.3 Physical sequence contraction

Carrier tokens reduce decoder cost only if the full resolution visual positions they represent are removed. For frame t, partition its n_t original visual positions into k_t consecutive segments of nearly equal size, and let $s_{t,j}$ be the final original decoder position in segment j. We keep all text positions and only these representative visual positions:

$$\mathcal{I} = \big(\{1, \ldots, S\} \setminus \mathcal{V}\big) \ \cup\ \{s_{t,j} : 1 \le t \le T,\ 1 \le j \le k_t\}\,. \tag{8}$$

Here $\mathcal{V}$ is the set of visual positions in the original prefix. We form $\widetilde{\mathbf{X}}$ by reading the positions in $\mathcal{I}$ in their original order, replacing the embedding at $s_{t,j}$ with carrier token $\mathbf{C}_t[j]$, and leaving every retained text embedding unchanged. Its length is

$$\widetilde{S} = S - n_{\mathrm{vis}} + \sum_{t=1}^{T} k_t. \tag{9}$$

The same selection applies to the attention mask, timestamps, and multimodal rotary position embeddings (MRoPE), so carrier token $\mathbf{C}_t[j]$ inherits the position of $s_{t,j}$ rather than a new one. Instantiated input lengths and compression ratios are in Section 4.

3.4 Video conditioned LoRA updates

The carrier tokens supply the decoder's explicit visual context. The hypernetwork adds a temporary weight space path that can use the full resolution features $\mathbf{F}_t$ absent from the contracted prefix. Let $\mathcal{A}$ denote the set of decoder layers with generated updates. For each $\ell \in \mathcal{A}$, let $\mathbf{H}^{(\ell)} \in \mathbb{R}^{\widetilde{S} \times d}$ be the residual stream at the input of layer ℓ, and let $\mathcal{P}_t$ be the positions of frame t's carrier tokens in the contracted prefix. The choice of $\mathcal{A}$ is specified in Section 4.

Conditioning vectors. The hypernetwork receives three conditioning vectors from each adapted layer: a carrier summary, a full resolution readout, and the final question state. The first averages the hidden states at every carrier token position. With $N_C = \sum_{t=1}^{T} k_t$ and $\mathcal{P} = \bigcup_t \mathcal{P}_t$,

$$\mathbf{v}^{(\ell)} = \frac{1}{N_C} \sum_{i \in \mathcal{P}} \mathbf{H}^{(\ell)}[i] \in \mathbb{R}^d. \tag{10}$$

Second, each carrier token state attends to the full resolution features of its own frame. Let $\mathbf{R}_t^{(\ell)} = \mathbf{H}^{(\ell)}[\mathcal{P}_t]$ be the $k_t \times d$ matrix of carrier token states for frame t. Using a shared multihead cross

attention block with residual connection,

$$\overline{\mathbf{R}}_t^{(\ell)} = \mathbf{R}_t^{(\ell)} + \mathrm{MHA}\Big(\mathrm{LN}(\mathbf{R}_t^{(\ell)}),\ \mathbf{F}_t,\ \mathbf{F}_t\Big), \qquad \mathbf{c}^{(\ell)} = \frac{1}{N_C}\sum_{t=1}^{T}\sum_{j=1}^{k_t}\overline{\mathbf{R}}_t^{(\ell)}[j]. \tag{11}$$

The MHA projections are trainable, and attention is independent across frames (frame t's carriers attend only to $\mathbf{F}_t$). The vector $\mathbf{c}^{(\ell)} \in \mathbb{R}^d$ is the mean full resolution readout across carrier tokens, and the factor $1/N_C$ keeps its scale independent of frame and carrier count.

Third, $\mathbf{q}^{(\ell)} = \mathbf{H}^{(\ell)}[p]$ is the state at the final question token position p, which attends to the preceding video and question under the causal mask. We concatenate the three,

$$\mathbf{g}^{(\ell)} = \Big[\mathbf{v}^{(\ell)}; \mathbf{c}^{(\ell)}; \mathbf{q}^{(\ell)}\Big] \in \mathbb{R}^{3d}. \tag{12}$$

Generating the update. The hypernetwork maps $\mathbf{g}^{(\ell)}$ to a latent vector $\mathbf{z}^{(\ell)} \in \mathbb{R}^{d_z}$. At each adapted layer, let m index a selected decoder projection with frozen weight $\mathbf{W}_m^{(\ell)} \in \mathbb{R}^{d_{\mathrm{out},m}\times d_{\mathrm{in},m}}$. For a chosen LoRA rank r, a module specific head $\mathbf{P}_m^{(\ell)} \in \mathbb{R}^{rd_z\times d_z}$ produces a normalized latent matrix, which two module specific projections $\mathbf{U}_m^{(\ell)} \in \mathbb{R}^{d_z\times d_{\mathrm{in},m}}$ and $\mathbf{V}_m^{(\ell)} \in \mathbb{R}^{d_z\times d_{\mathrm{out},m}}$ map to the two LoRA factors:

$$\mathbf{Z}_m^{(\ell)} = \mathrm{RowNorm}\Big(\mathrm{reshape}_{r\times d_z}\Big(\mathbf{P}_m^{(\ell)}\mathbf{z}^{(\ell)}\Big)\Big), \qquad \mathbf{A}_m^{(\ell)} = \mathbf{Z}_m^{(\ell)}\mathbf{U}_m^{(\ell)}, \qquad \mathbf{B}_m^{(\ell)} = \mathbf{Z}_m^{(\ell)}\mathbf{V}_m^{(\ell)}. \tag{13}$$

Here RowNorm normalizes each row to unit ℓ_2 norm, so $\mathbf{A}_m^{(\ell)} \in \mathbb{R}^{r\times d_{\mathrm{in},m}}$ and $\mathbf{B}_m^{(\ell)} \in \mathbb{R}^{r\times d_{\mathrm{out},m}}$, and the correction matches the frozen projection's shape:

$$\Delta\mathbf{W}_m^{(\ell)} = \frac{\alpha}{r}\Big(\mathbf{B}_m^{(\ell)}\Big)^{\top}\mathbf{A}_m^{(\ell)}, \tag{14}$$

with α the LoRA scale. Initializing each $\mathbf{V}_m^{(\ell)}$ to zero makes every $\Delta\mathbf{W}_m^{(\ell)}$ zero at initialization, so training begins from the frozen decoder on the contracted prefix. Architecture, rank, scale, adapted projections, and parameter count are in Section 4.

Prefill and decoding. During prefill, we apply the correction without forming a dense $d_{\mathrm{out},m} \times d_{\mathrm{in},m}$ matrix. For an input matrix $\mathbf{X}_{\mathrm{in}}$, the modified projection is

$$\mathrm{Linear}_m^{(\ell)}(\mathbf{X}_{\mathrm{in}}) = \mathbf{X}_{\mathrm{in}}\Big(\mathbf{W}_m^{(\ell)}\Big)^{\top} + \frac{\alpha}{r}\Big(\mathbf{X}_{\mathrm{in}}\Big(\mathbf{A}_m^{(\ell)}\Big)^{\top}\Big)\mathbf{B}_m^{(\ell)}. \tag{15}$$

The update depends only on the video question prefix, so it is constant across answer tokens. After prefill we merge the corrections into the frozen projections, decode with the ordinary frozen decoder over the shorter KV cache, and subtract them before the next example. The hypernetwork and low rank multiplication are never evaluated per generated token.

3.5 Training by context distillation

We train the compressor and hypernetworks by distilling from the frozen backbone (Hinton et al., 2015) on the uncompressed prefix. For each example the teacher computes logits on $\mathbf{X}$ without gradients and the student on $\widetilde{\mathbf{X}}$ with its generated updates (response positions are never contracted). With $\mathcal{R}$ the response token positions, $\mathbf{u}_r^{\mathrm{T}}, \mathbf{u}_r^{\mathrm{S}}$ the teacher and student logits at r, and y_r the answer token, define

$$\mathbf{p}_r^{\mathrm{T}} = \mathrm{softmax}\left(\frac{\mathbf{u}_r^{\mathrm{T}}}{\tau}\right), \qquad \mathbf{p}_r^{\mathrm{S}} = \mathrm{softmax}\left(\frac{\mathbf{u}_r^{\mathrm{S}}}{\tau}\right). \tag{16}$$

We minimize

$$\mathcal{L} = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\left[\tau^2\,\mathrm{KL}\big(\mathbf{p}_r^{\mathrm{T}}\,\big\|\,\mathbf{p}_r^{\mathrm{S}}\big) + \lambda\,\mathrm{CE}\big(\mathbf{u}_r^{\mathrm{S}}, y_r\big)\right]. \tag{17}$$

The KL term transfers the frozen model's full vocabulary distribution, and the cross entropy term anchors the student to the observed response.

Table 1: **Video question answering accuracy**, in percent. The short average covers the first four columns and the long average the next five. Rows labelled WIRE give our compressor and hypernetwork on a frozen Qwen3-VL-8B-Instruct backbone at four values of the per frame token budget k, all at 256 sampled frames. Bold marks the best entry among the efficient models re-evaluated under our controlled 256-frame protocol (WIRE, BIMBA, Vamba, VideoNSA). MLVU and LVBench entries are the *test* split. Table 4 in the appendix adds proprietary models, additional open baselines, and further efficient methods with the accuracies stated in their own papers.

Model	**NextQA** test (Xiao et al., 2021)	**PerceptionTest** test (Patraucean et al., 2023)	**MVBench** test (Li et al., 2024b)	**Tomato** test (Shangguan et al., 2024)	**Short QA avg.**	**Video-MME** test (Fu et al., 2025)	**Video-MME-Sub** test (Fu et al., 2025)	**LongVideoBench** val (Wu et al., 2024)	**MLVU** test MCQ (Zhou et al., 2025)	**LVBench** test (Wang et al., 2024)	**Long QA avg.**	**Average**
Open models												
Qwen3-VL-8B (Bai et al., 2025a)	83.4	72.7	68.7	35.7	65.1	71.4	75.2	62.4	57.6	58.0	64.9	65.0
Eagle2.5-8B (Chen et al., 2025)	85.0	81.0	74.8	31.0	68.0	72.4	75.7	66.4	60.4	50.9	65.2	66.4
PLM-8B (Cho et al., 2026)	84.1	82.7	77.1	33.2	69.3	58.3	65.4	56.9	52.6	44.5	55.5	61.6
LLaVA-Video-7B (Zhang et al., 2025d)	83.2	68.8	58.6	24.9	58.9	63.3	69.7	58.2	52.8	44.2	57.6	58.2
Molmo2-8B (Clark et al., 2026)	86.2	82.1	75.9	39.6	71.0	69.9	75.8	67.5	60.2	52.8	65.2	67.8
Efficient video language models (controlled re-evaluation at 256 frames)												
BIMBA (Islam et al., 2025)	80.9	65.8	59.6	24.0	57.6	63.3	66.4	57.7	**53.9**	43.3	56.9	57.2
Vamba (Ren et al., 2025)	77.5	**69.4**	59.6	26.2	58.2	58.1	61.2	52.8	45.7	41.2	51.8	54.6
VideoNSA (Song et al., 2026)	77.1	56.7	57.0	22.7	53.4	59.4	61.9	52.2	41.3	41.8	51.3	52.2
WIRE ($k = 16$)	**81.8**	66.7	**62.8**	**30.9**	**60.6**	**66.2**	**67.5**	**60.4**	52.0	**44.0**	**58.0**	**59.1**
WIRE ($k = 12$)	79.6	66.2	60.9	29.3	59.0	64.1	66.5	59.6	52.5	42.0	56.9	57.9
WIRE ($k = 8$)	80.0	65.4	60.6	28.7	58.7	63.8	66.3	58.8	51.9	40.0	56.2	57.3
WIRE ($k = 6$)	79.9	63.4	60.0	28.4	57.9	62.0	65.5	58.6	50.1	41.3	55.5	56.6

4 EXPERIMENTS

Architecture configuration. All experiments use Qwen3-VL-8B-Instruct as the frozen backbone (36 decoder layers of width 4,096). Neither its vision encoder nor its decoder is updated, and the fixed visual preprocessing produces $n_{\text{ref}} = 56$ full resolution visual tokens per frame. The compressor is one forward and one reverse Mamba2 block with state size 64. The hypernetwork uses a two layer trunk with latent width $d_z = 256$, adapts layers $\mathcal{A} = \{3, 7, \ldots, 35\}$ (every fourth layer), and generates rank-$r = 64$ updates with scale $\alpha = 128$ for the query, key, value, output, gate, up, and down projections. The trainable system totals 0.98B parameters (763M hypernetwork, 203M compressor, 16.8M shared cross attention).

Carrier token budgets and training. Unless stated otherwise, the primary model uses $k = 16$ carrier tokens per frame, compared with $n_{\text{ref}} = 56$ full resolution visual tokens per frame. We also train models with $k \in \{12, 8, 6\}$ to trace the accuracy versus efficiency frontier. We train on 1.65M multiple choice and open ended video question answering examples from LLaVA-Video-178K (Zhang et al., 2025d), with 128 sampled frames per training video. AdamW uses a peak learning rate of 3×10^{-5}, 3% warmup, cosine decay, bfloat16 arithmetic, and effective batch size 120. The distillation loss uses $\tau = 1$ and $\lambda = 0.1$.

Evaluation and paired update analysis. All efficient models are evaluated at 256 uniformly sampled frames with greedy decoding on NextQA, PerceptionTest, MVBench, TOMATO, Video-MME with and without subtitles, LongVideoBench, MLVU, and LVBench (Xiao et al., 2021; Patraucean et al., 2023; Li et al., 2024b; Shangguan et al., 2024; Fu et al., 2025; Wu et al., 2024; Zhou et al., 2025; Wang et al., 2024). WIRE trains at 128 frames but no component is retrained or rescaled for longer clips. The carrier budget scaling (Eq. (4)) and inherited multimodal rotary position embeddings (MRoPE) are frame count agnostic. We rerun BIMBA, Vamba, and VideoNSA under the same protocol within their supported frame ranges. Latency is on one B200 GPU at batch size one in bfloat16, with the full grid in Table 5.

5 RESULTS

5.1 WIRE PROVIDES THE STRONGEST COMPLETE COVERAGE EFFICIENT RESULT

Table 1 compares WIRE against efficient video llms we reevaluate under a common 256 frame budget. We match frame count, decoding, hardware, and evaluation suite, while backbone, training

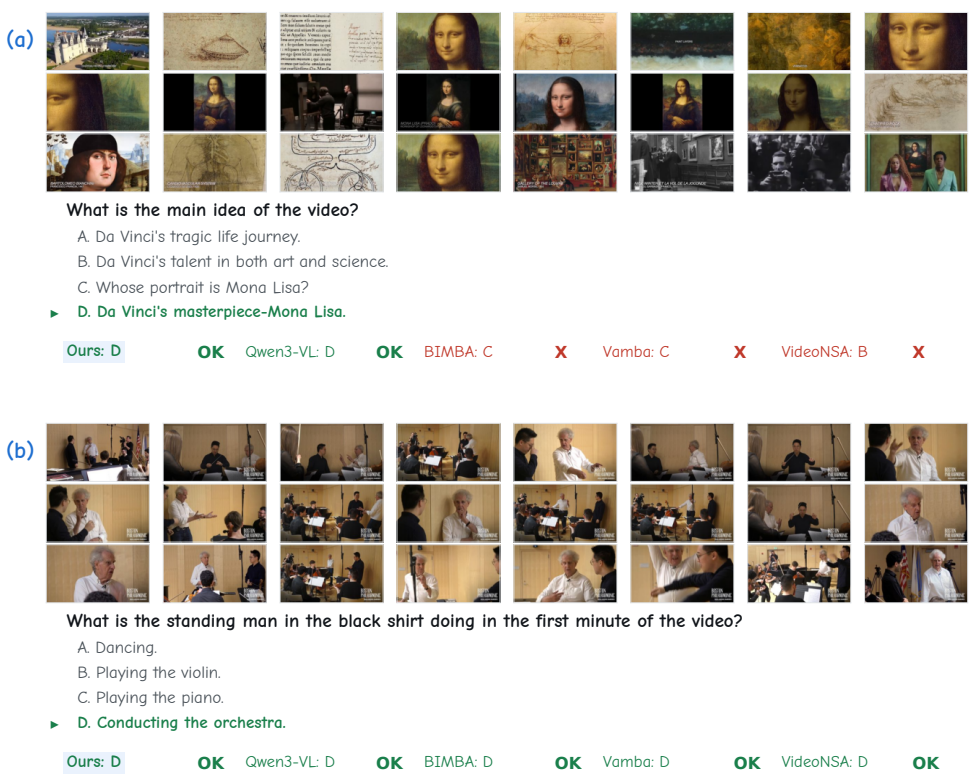


Figure 3: **Qualitative examples at 256 frames.** On a question about a video's global subject, WIRE matches the uncompressed backbone while the efficient baselines pick locally plausible distractors. On a temporally localized question all systems answer correctly.

data, and architecture stay method specific. This is a matched evaluation budget, not a controlled architectural comparison. Among these rows, WIRE with $k = 16$ carrier tokens attains the highest short, long, and overall averages, leading the short form, multimodal, subtitle aware, and long video evaluations at once. It stays competitive down to $k = 6$. It trails only on two columns: Vamba on PerceptionTest, whose recurrent cross attention keeps every video token, and BIMBA on MLVU.

Table 4 lists further efficient models at their reported accuracies. With different backbones, and protocols they are not directly comparable, so we neither bold nor aggregate them, but on the benchmarks each reports, WIRE is competitive with or ahead of them. These methods span the design space around WIRE (Ren et al., 2025; Song et al., 2026; Shen et al., 2025; Zohar et al., 2025; Tao et al., 2025; Jiang et al., 2025; Ding et al., 2026; Kim et al., 2026) but all keep their evidence in the activation stream, whereas WIRE contracts it and moves the missing evidence into a weight update. WIRE trades part of that visual context for a much shorter decoder sequence, accuracy declining smoothly from $k = 16$ to $k = 6$. Appendix A details every baseline and Figure 3 gives examples.

5.2 THE GENERATED UPDATE DOES TASK-RELEVANT WORK

The paired update analysis in Table 2 isolates the update's contribution: each checkpoint is run twice, toggling only the generated update while the video, question, carrier tokens, prompt, and decoding are held fixed (exact protocol in Appendix B). Any accuracy difference is therefore attributable to the generated update alone. Applying it raises whole benchmark accuracy by $+3.96$ on Video-MME, $+3.22$ on MVBench, and $+4.39$ on LongVideoBench. The subset breakdown, organized *a priori* around the capabilities that compression should strain, shows the largest gains on exactly those subsets (counting, temporal action, counterfactual inference, and subtitle grounding), reaching $+12.32$ points on LongVideoBench subtitle grounding. The update is most valuable precisely where the carrier tokens lose the most, indicating that it restores evidence removed by contraction rather than applying a generic answer calibration.

Table 2: **Paired update analysis at fixed compressed context.** Disc. discards the per example LoRA update and Appl. applies it; Δ is the paired difference in points at identical carrier context. † marks subsets not significant under the exact McNemar test.

Subset	Disc.	Appl.	Δ	Subset	Disc.	Appl.	Δ
Video-MME				*LongVideoBench*			
Counting	40.67	44.03	+3.36	Sub. → next	45.21	57.53	+12.32
Law	61.19	67.62	+6.43	Scene → sub.†	32.88	43.84	+10.96
Athletics	54.36	59.23	+4.87	Sub. → object†	55.41	66.22	+10.81
MVBench				*MLVU*			
Action count	46.33	51.83	+5.50	Count	28.33	33.33	+5.00
Counterfactual	51.67	56.33	+4.66				

Table 3: Compressor ablations at k=16 on Video-MME without subtitles (accuracy %, Δ vs. the full compressor). Each variant changes only the compressor.

Compressor variant	Video-MME (no sub.)	Δ
WIRE (full compressor)	66.2	—
− question conditioning (test time)	64.3	−1.9
− bidirectional scan (forward only, retrained)	65.1	−1.1
− selective scan (bilinear pooling only, retrained)	64.7	−1.5

5.3 COMPRESSOR ABLATIONS

We ablate the conditioned bidirectional selective scan of Section 3.2, holding WIRE fixed at k=16 on Video-MME without subtitles (full compressor 66.2, Table 3). Each variant changes only the compressor, so the differences isolate how the carrier tokens are formed. The largest effect comes from question conditioning: removing the question from the scan input at test time drops accuracy to 64.3 (−1.9). At k=16 each carrier token stands in for 3.5 full resolution tokens, so the compressor cannot keep everything; the drop shows it exploits the question to decide what the carrier tokens retain. A retrained forward only scan reaches 65.1 (−1.1): with one direction, early carrier tokens never see later content or the trailing question, whereas the reverse scan with $\mathbf{Q}$ at both ends (Eq. (6)) puts every carrier token near the question in at least one direction. Replacing the scan with bilinear pooling alone (Eq. (5)) gives 64.7 (−1.5): the parameter free initialization is a strong baseline but purely local, each carrier token a fixed combination of adjacent tokens in its own frame, so the recovered +1.5 is the value of learned, question aware mixing across the whole sequence. A needle in a haystack probe of what the compressed context preserves across carrier budgets is given in Appendix C.

6 DISCUSSION

WIRE splits video-conditioned computation between a short explicit carrier-token context and a temporary, video-conditioned weight update generated by hypernetworks and merged after prefill. At 16 carrier tokens per frame, it contracts the visual stream by 3.5×, reducing decoder prefill time by 3.35× at 256 frames while maintaining near-constant autoregressive decoding latency and proportionally reducing KV-cache usage. Under a matched 256-frame budget, WIRE achieves the strongest average accuracy among the efficient video models evaluated across nine benchmark settings. Moreover, paired ablations that hold the compressed visual context fixed show that the generated weight update performs task-relevant computation beyond the information retained by the carrier tokens, with gains of up to 4.4 points on established video understanding benchmarks.

Limitations Several claims are scoped by present evidence. The paired update does task relevant work, but its video specificity remains a hypothesis awaiting the mismatched video control of Appendix B. WIRE uses a single frozen backbone (Qwen3-VL-8B) and public benchmarks, so its generality is untested; the hypernetwork adds prefill compute, so savings appear only when generation is long relative to the video; and the main comparisons report a matched evaluation budget rather than a controlled architectural comparison.

ETHICS STATEMENT

WIRE builds on a frozen public backbone and public video question answering benchmarks and introduces no new data collection or human subjects. It inherits the biases and failure modes of its backbone. Cheaper long video understanding can aid accessibility and analysis but can also lower the cost of harmful large scale video surveillance or profiling; we intend the method for research and beneficial use.

REPRODUCIBILITY STATEMENT

Section 3 gives the full method and objective, Section 4 lists the backbone, carrier budgets, adapted layers, ranks, optimizer, and training data, and Appendix B details the selective scan recurrence, adapter head shapes, DeepStack alignment, two pass training, and the exact paired update and latency protocols. All benchmarks and backbones are public, so the reported results can be reproduced from these specifications.

USE OF AI ASSISTANTS

AI based tools were used only to polish grammar, wording, and phrasing of the manuscript. They were not used to generate research ideas, design or run experiments, produce results, or draw conclusions, all of which are the authors' own work.

REFERENCES

M. H. I. Abdalla, Zhipin Wang, Christian Frey, Steffen Eger, and Josif Grabocka. Zhyper: Factorized hypernetworks for conditioned LLM fine-tuning, 2026. URL https://openreview.net/forum?id=p0hiJ09B2V.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=EbMuimAbPbs.

Anthropic. Claude sonnet 4.5 system card, 2025. URL https://assets.anthropic.com/m/12f214efcc2f457a/original/Claude-Sonnet-4-5-System-Card.pdf.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Jimmy Ba, Geoffrey Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4338–4346, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

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.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. *Proceedings of the International Conference on Learning Representation*, 2025.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-lora: Instant transformer adaption. *arXiv preprint arXiv:2506.06105*, 2025.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Tjarko Lange. Doc-to-lora: Learning to instantly internalize contexts. *arXiv preprint arXiv:2602.15902*, 2026.

Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Tuomas Rintamaki, Tyler Poon, Max Ehrlich, Tuomas Rintamaki, Tyler Poon, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models, 2025. URL https://arxiv.org/abs/2504.15271.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXI*, 2024.

Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. Scaling video-language models to 10k frames via hierarchical differential distillation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=1CK1kuH1he.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476.

Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. Perceptionlm: Open-access data and models for detailed visual understanding. *Advances in Neural Information Processing Systems*, 38:23475–23537, 2026.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026. URL https://arxiv.org/abs/2601.10611.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. 2024.

Yang Ding, Xin Lai, Yizhen Zhang, Wei Li, Ruihang Chu, and Yujiu Yang. Videozoomer: Reinforcement-learned temporal focusing for long video reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ARHCFvgx6G.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Atri Rudra, James Y Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study. In *International Conference on Learning Representations*, volume 2026, pp. 42642–42687, 2026.

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, 2025. URL https://arxiv.org/abs/2405.21075.

Google. Gemini 3 Pro model card, 2025. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022. URL https://arxiv.org/abs/2111.00396.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1609.09106.

Xiaolong Han, Zehong Wang, Bo Zhao, Binchi Zhang, Jundong Li, Damian Borth, Rose Yu, Haggai Maron, Yanfang Ye, Lu Yin, and Ferrante Neri. A survey of weight space learning: Understanding, representation, and generation, 2026. URL https://arxiv.org/abs/2603.10090.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria, July 2025. Association for Computational Linguistics.

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

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention, 2021. URL https://arxiv.org/abs/2103.03206.

Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Storm: Token-efficient long video understanding for multimodal llms. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 5889–5900. IEEE, 2025.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.47. URL https://aclanthology.org/2021.acl-long.47/.

Dahye Kim, Bhuvan Sachdeva, Karan Uppal, Naman Gupta, Vineeth N. Balasubramanian, and Deepti Ghadiyaram. Swift sampling: Selecting temporal surprises via taylor series, 2026. URL https://arxiv.org/abs/2605.22678.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. Infinipot-v: Memory-constrained KV cache compression for streaming video understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=hFxOZjHyTg.

Aakash Sunil Lahoti, Kevin Li, Berlin Chen, Caitlin Wang, Aviv Bick, Zico Kolter, Tri Dao, and Albert Gu. Mamba-3: Improved sequence modeling using state space principles. In *International Conference on Learning Representations*, volume 2026, pp. 86173–86202, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 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 *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXVI*, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2024b. URL https://arxiv.org/abs/2311.17005.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

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, Hongyuan Tao, Qian Zhang, Tianheng Cheng, Yingyue Li, Haoran Yin, Wenyu Liu, and Xinggang Wang. Multimodal mamba: Decoder-only multimodal state space model via quadratic to linear distillation. *arXiv preprint arXiv:2502.13145*, 2025.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024a.

Xinxin Liu, Shiwei Gan, Xiao Liu, Yafeng Yin, Lei Xie, and Sanglu Lu. Infomerge: Information-aware token compression for efficient video large language models, 2026a. URL https://arxiv.org/abs/2606.02161.

Yewei Liu, Xiyuan Wang, Yansheng Mao, Yoav Gelberg, Haggai Maron, and Muhan Zhang. SHINE: A scalable in-context hypernetwork for mapping context to loRA in a single pass. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=ZMexYcAibv.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Vishwesh Nath, Jinyi Hu, Sifei Liu, Ranjay Krishna, Daguang Xu, Xiaolong Wang, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. Nvila: Efficient frontier visual language models, 2024b. URL https://arxiv.org/abs/2412.04468.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

OpenAI. GPT-5 system card, 2025. URL https://openai.com/index/gpt-5-system-card/.

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens Continente, Larisa Markeeva, Dylan Sunil 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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=HYEGXFnPoq.

Aleksandar Petrov, Mark Sandler, Andrey Zhmoginov, Nolan Miller, and Max Vladymyrov. Long context in-context compression by getting to the gist of gisting. *arXiv preprint arXiv:2504.08934*, 2025.

Minghao Qin, Xiangrui Liu, Zhengyang Liang, Yan Shu, Huaying Yuan, Juenjie Zhou, Shitao Xiao, Bo Zhao, and Zheng Liu. Video-xl-2: Towards very long-video understanding through task-aware kv sparsification, 2025. URL https://arxiv.org/abs/2506.19225.

Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. Vamba: Understanding hour-long videos with hybrid mamba-transformers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21197–21208. IEEE, 2025.

Jürgen Schmidhuber. Learning to control fast-weight memories: an alternative to dynamic recurrent networks. *Neural Comput.*, 4(1):131–139, January 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.131. URL https://doi.org/10.1162/neco.1992.4.1.131.

Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models, 2024. URL https://arxiv.org/abs/2410.23266.

Kele Shao, Keda TAO, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=G2od9JVHkE. Survey Certification.

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

Haebin Shin, Lei Ji, Yeyun Gong, Sungdong Kim, Eunbi Choi, and Minjoon Seo. Generative prompt internalization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7338–7363, 2025.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Enxin Song, Wenhao Chai, Shusheng Yang, Ethan J. Armand, Xiaojun Shan, Haiyang Xu, Jianwen Xie, and Zhuowen Tu. VideoNSA: Native sparse attention scales video understanding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zA2LbsUMDd.

Zhaoxuan Tan, Zixuan Zhang, Haoyang Wen, Zheng Li, Rongzhi Zhang, Pei Chen, Fengran Mo, Zheyuan Liu, Qingkai Zeng, Qingyu Yin, and Meng Jiang. Instant personalized large language model adaptation via hypernetwork. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23557–23580, San Diego, California, United States, July 2026. Association for Computational Linguistics.

Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. *arXiv preprint arXiv:2502.21271*, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Haochen Li, Jiale Zhu, Jiali Chen, Jiaxing Xu, Jiazheng Xu, Jing Chen, Jinghao Lin, Jinhao Chen, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Ruiliang Lyu, Shangqin Tu, Sheng Yang, Shengbiao Meng, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wei Jia, Wenkai Li, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyu Zhang, Xinyue Fan, Xuancheng Huang, Yadong Xue, Yanfeng Wang, Yanling Wang, Yanzi Wang, Yifan An, Yifan Du, Yiheng Huang, Yilin Niu, Yiming Shi, Yu Wang, Yuan Wang, Yuanchang Yue, Yuchen Li, Yusen Liu, Yutao Zhang, Yuting Wang, Yuxuan Zhang, Zhao Xue, Zhengxiao Du, Zhenyu Hou, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026. URL https://arxiv.org/abs/2507.01006.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HMrcv7Q4Ub.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Xiao Wang, Qingyi Si, Shiyu Zhu, Jianlong Wu, Li Cao, and Liqiang Nie. AdaReTaKe: Adaptive redundancy reduction to perceive longer for video-language understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5417–5432, Vienna, Austria, July 2025b. Association for Computational Linguistics.

Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, Min Dou, Kai Chen, Wenhai Wang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2.5: Empowering video mllms with long and rich context modeling, 2025c. URL https://arxiv.org/abs/2501.12386.

Zixuan Wang, Xingyu Dang, Rui-Jie Zhu, Zixin Wen, Hengyu Fu, Wenhao Chai, and Jason D. Lee. Learning what to remember: Test-time training via context distillation, 2026. URL https://arxiv.org/abs/2608.01672.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv*, 2023.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9777–9786, June 2021.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction, 2025. URL https://arxiv.org/abs/2410.17247.

Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-owl3: Towards long image-sequence understanding in multi-modal large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=pr37sbuhVa.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 29836–29846, June 2025b.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11704–11715, 2026.

Boya Zeng, Yida Yin, Zhiqiu Xu, and Zhuang Liu. Generative modeling of weights: Generalization or memorization?, 2025. URL https://openreview.net/forum?id=QbSeApdgfi.

Yiming Zeng, Jinghan Cao, Zexin Li, Wanhao Yu, Zhankai Ye, Dawei Xiang, Ting Hua, Xin Liu, Shangqian Gao, and Tingting Yu. Hyperedit: Unlocking instruction-based text editing in llms via hypernetworks. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 466–480, 2026a.

Yiming Zeng, Lei Lu, Zexin Li, Zhuochun Li, Shuoqiu Li, Shuyi Liao, Xidong Wu, Zeyu Zhang, Minmei Wang, Yu Zhao, Tingting Yu, and Shangqian Gao. Moegen: Mixture-of-experts for instance-adaptive lora generation, 2026b. URL https://arxiv.org/abs/2608.03275.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. Videollama 3: Frontier multimodal foundation models for image and video understanding, 2025a. URL https://arxiv.org/abs/2501.13106.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token, 2025b. URL https://arxiv.org/abs/2501.03895.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*, 2025c. URL https://openreview.net/forum?id=80faIPZ67S.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2025d. URL https://arxiv.org/abs/2410.02713.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: heavy-hitter oracle for efficient generative inference of large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zheyu Aqa Zhang, Ziqi Pang, Shixing Chen, Xiang Hao, Vimal Bhat, and Yu-Xiong Wang. One token per highly selective frame: Towards extreme compression for long video understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025e.

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, 2025. URL https://arxiv.org/abs/2406.04264.

Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18891–18901. IEEE, 2025.

A BASELINE IMPLEMENTATIONS AND COMPARISON SCOPE

Comparison scope. We group the efficient models into a matched budget set that we reevaluate at 256 uniformly sampled frames under one protocol (BIMBA, Vamba, VideoNSA, and WIRE) and a reported set whose scores are copied from each method's own paper. The matched budget set fixes the evaluation budget (frame count, decoding, hardware, and suite), so these rows isolate an evaluation budget rather than the compression mechanism. Architecture only conclusions are therefore not warranted across every row, because checkpoints, vision and language backbones, training mixtures, sampling rules, spatial resolution, visual token budgets, prompting, and decoding can differ. For the reported set we take only the MLVU and LVBench *test* splits and leave a cell blank when a paper reports only the MLVU dev or M-avg number, and we compute aggregate scores only for rows that cover the complete suite. Several concurrent open video language models (InternVideo2.5 (Wang et al., 2025c), MiniCPM-V 4.5 (Yu et al., 2026), and InternVL3.5 (Wang et al., 2025a)) report on overlapping suites but under their own frame budgets, backbones, and training mixtures, so we describe rather than tabulate them as budget matched rows. The following descriptions identify the mechanism represented by each baseline and its principal comparison qualifier.

A.1 OPEN REFERENCE MODELS

Qwen3-VL-8B. Qwen3-VL combines a SigLIP-2 vision encoder, a two layer vision language merger, and a Qwen3 decoder (Bai et al., 2025a). Interleaved MRoPE, DeepStack visual injection, and timestamp text support multimodal temporal representation. It is an open full context reference rather than a long video compression method. The 8B release includes Instruct and Thinking variants, so a comparison must specify the checkpoint, reasoning mode, frame sampling, visual budget, and prompting protocol.

Eagle2.5-8B. Eagle 2.5 projects SigLIP visual features into a Qwen2.5 language model through an MLP and uses LLaVA style image tiling (Chen et al., 2025). Its Image Area Preservation and Automatic Degradation Sampling preserve aspect ratio and allocate the input length budget between text and visual tokens. It is a long context reference rather than a temporal compressor. Reported Video-MME settings include 512 frames, so backbone, frame count, and resolution differ from a fixed 256 frame comparison.

PerceptionLM (PLM-8B). PerceptionLM links a pretrained Perception Encoder to a Llama 3 decoder using a two layer MLP projector (Cho et al., 2026). Its 8B variant uses a PE G/14 encoder and Llama 3.1 8B. For video, it retains sampled frames and applies spatial pooling within each frame rather than selecting a shorter temporal sequence. Its visual backbone, resolution, and frame protocol differ from the efficient model configurations here.

LLaVA-Video-7B. LLaVA-Video extends LLaVA-OneVision with synthetic video instruction data and uses SigLIP with a Qwen2 language model (Zhang et al., 2025d). Its optional SlowFast representation allocates different spatial pooling rates to slow and fast frame groups, allowing broader temporal coverage at a bounded token count. Several published benchmark results are marked for overlap between a corresponding dataset and the training mixture, so such scores are not strictly independent comparisons.

LLaVA-OneVision-7B. LLaVA-OneVision is an open full context model on a Qwen2-7B backbone that unifies single image, multi image, and video transfer (Li et al., 2025). Each frame is token reduced by bilinear interpolation and evaluated at 32 frames. It is an open reference rather than a token compression method, and it does not report TOMATO, MLVU test, or LVBench.

Molmo2-8B. Molmo2 encodes independently sampled video frames with a SigLIP-2 style vision transformer and connector, spatially pools video patches, and interleaves the resulting visual tokens with timestamp text (Clark et al., 2026). It uses pretraining, multitask supervised finetuning, and a long context stage. The reported video protocol uses fixed sampling and frame caps, which materially affect context length and compute. It is consequently an open reference, not an architecture matched token compression baseline.

VideoChat2. VideoChat2 couples a UMT-L visual encoder and a QFormer with a 7B language backbone and is the reference model introduced with the MVBench benchmark (Li et al., 2024b). It is widely retabulated in later long video evaluations. We report its Mistral-7B numbers under uniform 16 frame sampling and use its zero shot scores, so its MVBench value is comparatively strong while its long video Video-MME scores are modest.

A.2 Efficient video language models

BIMBA. BIMBA jointly processes frame derived visual features and a smaller set of visual queries, interleaves those queries through the sequence, and uses a bidirectional selective scan before sending compressed tokens to an LLM (Islam et al., 2025). This reduces the LLM facing context while replacing full sequence attention in the compressor with linear time sequence processing. Its paper includes a shared data, shared decoder comparison, but cross paper rows can otherwise differ in both decoder and training data.

Vamba. Vamba avoids visual token reduction. It retains transformer self attention for text, replaces video token causal self attention with Mamba2, and connects modalities through text query/video key value cross attention (Ren et al., 2025). Its efficiency comes from linear recurrent processing over video tokens rather than a shorter visual prefix. The reported runtime comparison uses a shared Qwen2-VL setup, but Vamba adds substantial parameters and uses different training and inference frame capacities.

VideoNSA. VideoNSA applies learnable Native Sparse Attention to vision tokens in Qwen2.5-VL-7B (Bai et al., 2025b) while retaining standard grouped query attention for text (Song et al., 2026). Gated branches compress blocks, select salient blocks, and retain a sliding local window, reducing vision key value interactions without all to all visual attention. Its main model is end to end finetuned on a filtered LLaVA-Video subset, a training condition that should be kept separate from methods evaluated with their released configurations.

B Implementation details

Selective scan recurrence. The state space layer maintains a state $\mathbf{h}_i \in \mathbb{R}^{N\times d'}$, where $N = 64$ is the SSM state size and d' is the per channel model width, taken equal to the hidden width d up to the Mamba2 expansion factor. Writing z_i for the i-th row of the scan input $\mathbf{Z}$ (distinct from the hypernetwork latent $\mathbf{z}^{(\ell)}$ of Eq. (12)), the state updates with input dependent gates,

$$\mathbf{h}_i = \bar{A}(z_i) \odot \mathbf{h}_{i-1} + \bar{B}(z_i)\, z_i^\top, \qquad y_i = C(z_i)^\top \mathbf{h}_i + D \odot z_i, \tag{18}$$

where $\bar{B}(z_i) \in \mathbb{R}^N$, so that $\bar{B}(z_i)\, z_i^\top$ is the rank one injection of the input into the state. We use the state space duality Mamba2 formulation with state size $N = 64$ and separate parameters for the forward and reverse scans. The scan output projection is initialized from $\mathcal{N}(0, (3\times 10^{-4})^2)$, which preserves the bilinear resampling initialization while allowing gradients to reach the scan interior from the first step.

Adapter head shapes. Mirroring Eq. (13), the module specific projection $\mathbf{P}_m^{(\ell)} \in \mathbb{R}^{rd_z\times d_z}$ reshapes the module latent into $\mathbf{Z}_m^{(\ell)} \in \mathbb{R}^{r\times d_z}$, and the mixing matrices $\mathbf{U}_m^{(\ell)} \in \mathbb{R}^{d_z\times d_{\mathrm{in},m}}$ and $\mathbf{V}_m^{(\ell)} \in \mathbb{R}^{d_z\times d_{\mathrm{out},m}}$ map it to the LoRA factors $\mathbf{A}_m^{(\ell)}$ and $\mathbf{B}_m^{(\ell)}$. The $\mathrm{RowNorm}$ operator normalizes $\mathbf{Z}_m^{(\ell)}$ row by row before the mixing step, keeping the generated rank one directions at a stable scale.

DeepStack alignment. Qwen3-VL injects visual encoder features at several early decoder layers. We pool these features using the same segment map as Eq. (8), keeping every DeepStack feature aligned with the retained token that represents its segment.

Efficient two pass training. The vocabulary has 151,936 entries, so full sequence teacher and student logits are not materialized. Instead, both passes return final hidden states and the frozen output head forms logits only at response positions, in chunks. Teacher and student use identical

pixels, and because the vision encoder is frozen, its features are computed once and reused by the student in detached form. Under activation checkpointing, the student forward must remain the final forward before backward so recomputation retrieves the correct contraction state.

Paired update analysis protocol. For every evaluation question in Table 2, the compressor constructs one contracted decoder input from the same sampled video frames and prompt. This input includes the same carrier tokens in both conditions. We then run the same frozen decoder twice: once with the LoRA update produced by the hypernetworks and once with that update discarded. The response parser and decoding rule are fixed. The paired accuracy difference therefore measures what the generated update adds on top of the same compressed visual context.

Latency measurements. Table 5 reports prefill, time to first token, and per token decode latency for the base backbone and WIRE at $k \in \{6, 16\}$ across 128 to 1024 frames. Prefill falls by $3.35\times$ at 256 frames and $4.14\times$ at 1024 frames, while per token decode latency is essentially unchanged because the generated update is merged into the frozen weights before decoding.

Table 4: **Video question answering accuracy, full comparison**, in percent. Table 1 in the main text is the subset of these rows most directly comparable to WIRE. As in Table 1, the final efficient block reports accuracies stated in each method's own paper under different backbones, frame counts, and protocols, so these rows are partial and receive no aggregate.

Model	**NextQA** test (Xiao et al., 2021)	**PerceptionTest** test (Patraucean et al., 2023)	**MVBench** test (Li et al., 2024b)	**Tomato** test (Shangguan et al., 2024)	**Short QA avg.**	**Video-MME** test (Fu et al., 2025)	**Video-MME-Sub** test (Fu et al., 2025)	**LongVideoBench** val (Wu et al., 2024)	**MLVU** test MCQ (Zhou et al., 2025)	**LVBench** test (Wang et al., 2024)	**Long QA avg.**	**Average**
Proprietary models, API only												
GPT-5 (OpenAI, 2025)	86.3	79.4	74.1	53.0	73.2	83.3	86.9	72.6	77.7	65.2	77.1	75.4
GPT-5 mini (OpenAI, 2025)	83.2	72.0	66.5	44.1	66.5	77.3	82.3	69.7	69.1	54.7	70.6	68.8
Gemini 3 Pro (Google, 2025)	84.3	77.6	70.4	48.3	70.2	88.6	87.5	75.9	75.7	77.0	80.9	76.1
Gemini 2.5 Pro (Comanici et al., 2025)	85.3	78.4	70.6	48.6	70.7	87.8	87.8	76.8	81.5	75.7	81.9	76.9
Gemini 2.5 Flash (Comanici et al., 2025)	81.8	74.7	67.0	39.1	65.7	84.2	84.2	73.1	75.1	64.9	76.3	71.6
Claude Sonnet 4.5 (Anthropic, 2025)	79.2	64.3	62.1	39.6	61.3	74.2	80.5	65.1	64.0	50.5	66.9	64.4
Open models												
InternVL3.5-4B (Wang et al., 2025a)	80.3	68.1	71.2	26.8	61.6	65.4	68.6	60.8	52.0	43.2	58.0	59.6
InternVL3.5-8B (Wang et al., 2025a)	81.7	72.7	72.1	24.6	62.8	66.0	68.6	62.1	53.2	43.4	58.7	60.5
Qwen3-VL-4B (Bai et al., 2025a)	81.4	70.7	68.9	31.8	63.2	69.3	74.0	62.8	58.4	56.2	64.1	63.7
Qwen3-VL-8B (Bai et al., 2025a)	83.4	72.7	68.7	35.7	65.1	71.4	75.2	62.4	57.6	58.0	64.9	65.0
Keye-VL-1.5-8B (Yang et al., 2025)	75.8	64.2	56.9	33.0	57.5	73.0	76.2	66.0	53.8	42.8	62.4	60.2
GLM-4.1V-9B (Team et al., 2026)	81.3	74.2	68.4	30.0	63.5	68.2	75.6	65.7	56.6	44.0	62.0	62.7
MiniCPM-V-4.5-8B (Yu et al., 2026)	78.8	70.9	60.5	29.8	60.0	67.9	73.5	63.9	60.6	50.4	63.3	61.8
Eagle2.5-8B (Chen et al., 2025)	85.0	81.0	74.8	31.0	68.0	72.4	75.7	66.4	60.4	50.9	65.2	66.4
PLM-3B (Cho et al., 2026)	83.4	79.3	74.7	30.9	67.1	54.9	59.4	57.9	48.4	40.4	52.2	58.8
PLM-8B (Cho et al., 2026)	84.1	82.7	77.1	33.2	69.3	58.3	65.4	56.9	52.6	44.5	55.5	61.6
LLaVA-Video-7B (Zhang et al., 2025d)	83.2	68.8	58.6	24.9	58.9	63.3	69.7	58.2	52.8	44.2	57.6	58.2
LLaVA-OneVision-7B (Li et al., 2025)	79.4	57.1	56.7	–	–	58.2	61.5	56.4	–	–	–	–
VideoChat-Flash-7B (Li et al., 2026)	85.5	76.5	74.0	32.5	67.1	65.3	69.7	64.7	56.0	48.2	60.8	63.6
Molmo2-4B (Clark et al., 2026)	85.5	81.3	75.1	39.8	70.4	69.6	75.7	68.0	63.0	53.9	66.0	68.0
Molmo2-8B (Clark et al., 2026)	86.2	82.1	75.9	39.6	71.0	69.9	75.8	67.5	60.2	52.8	65.2	67.8
Molmo2-O-7B (Clark et al., 2026)	84.3	79.6	74.8	36.2	68.7	64.9	69.2	63.7	55.2	49.6	60.5	64.2
Efficient video language models (re-evaluated at a common 256-frame budget)												
BIMBA (Islam et al., 2025)	80.9	65.8	59.6	24.0	57.6	63.3	66.4	57.7	**53.9**	43.3	56.9	57.2
Vamba (Ren et al., 2025)	77.5	**69.4**	59.6	26.2	58.2	58.1	61.2	52.8	45.7	41.2	51.8	54.6
VideoNSA (Song et al., 2026)	77.1	56.7	57.0	22.7	53.4	59.4	61.9	52.2	41.3	41.8	51.3	52.2
WIRE ($k = 16$)	**81.8**	66.7	**62.8**	**30.9**	**60.6**	**66.2**	**67.5**	**60.4**	52.0	**44.0**	**58.0**	**59.1**
WIRE ($k = 12$)	79.6	66.2	60.9	29.3	59.0	64.1	66.5	59.6	52.5	42.0	56.9	57.9
WIRE ($k = 8$)	80.0	65.4	60.6	28.7	58.7	63.8	66.3	58.8	51.9	40.0	56.2	57.3
WIRE ($k = 6$)	79.9	63.4	60.0	28.4	57.9	62.0	65.5	58.6	50.1	41.3	55.5	56.6
Additional video language models, scores as reported by their original papers (mixed backbones and protocols, partial coverage)												
LLaVA-Mini (Zhang et al., 2025b)	–	–	44.5	–	–	–	–	–	–	–	–	–

Table 5: Latency of the uncompressed Qwen3-VL-8B backbone and WIRE on one B200 GPU in bfloat16. All timings are batch size one: per example updates cannot be batched across examples. For WIRE, prefill and time to first token include the compressor, the hypernetwork forward, and the cost of merging the generated update. Prefill excludes the vision encoder, while time to first token includes it. Decoding is measured after merging the generated update.

		prefill (ms)		TTFT (ms)			decode (ms/token)	
k	frames	base	WIRE	base	WIRE	encoder	base	WIRE
16	128	180	54	274	162	94	18.8	19.7
16	256	369	110	550	323	181	18.9	18.8
16	512	851	228	1216	659	365	21.1	18.9
16	1024	2086	504	2819	1367	733	21.1	18.2
6	128	183	30	276	136	93	19.0	19.1
6	256	359	57	550	273	192	18.5	19.2
6	512	824	111	1199	542	375	20.6	21.9
6	1024	2152	238	2900	1099	748	20.0	19.1

C WHAT THE GENERATED UPDATE ENCODES

The ablations of Section 5.2 bound the contribution of the generated low rank update on benchmark questions, where a large part of each answer is recoverable from the question text alone. To measure

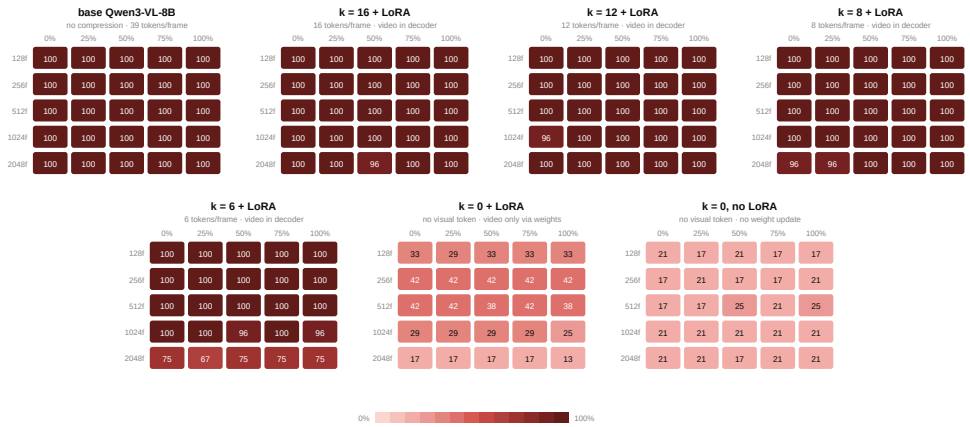


Figure 4: **Video needle in a haystack under decoder input ablation.** Retrieval accuracy (%) for a planted word card needle across five insertion depths (columns of each grid) and three haystack lengths (rows). Darker cells are higher. The uncompressed backbone and WIRE at carrier budgets $k \in \{16, 12, 8, 6\}$ keep the video in the decoder input. The two k=0 grids remove every visual token, so the generated update is the only path from video to answer, and differ only in whether that update is applied. Chance is 25%.

the update's information content without that component, we adapt the needle in a haystack protocol of Cheng et al. (2025) so that the answer is a target the question text cannot supply.

Protocol. A needle word (drawn from a 32 word vocabulary) is rendered across 32 consecutive frames and spliced into a LongVideoBench haystack at one of five depths (0 to 100%), then probed as a four way multiple choice. The word is determined by neither haystack nor question, so chance is 25%. Haystacks run 256 to 2048 frames (the needle falls from 12.5% to 1.6% of the input), 120 items per model and length. Options are rotated to a uniform key and the per frame resolution is fixed at 79,872 px, so letter bias and shrinking frames do not confound the measurement.

Contraction preserves the needle, the generated update does not carry it. Figure 4 separates the two paths by which the video can reach the answer. With the contracted video present in the decoder input sequence, every width from k=16 down to k=8 matches the uncompressed model at every position and every sequence length, at $8/39$ of the visual tokens. The k=6 width is the first to lose the needle. It holds through 1024 frames and falls to 73.3 at 2048. Accuracy is flat across the five positions for every model, so the loss is a function of the fraction of the input the needle occupies and not of where it sits.

The k=0 checkpoint, trained with no visual token in the decoder input sequence, isolates the other path: the generated update is the only route from the video to the output, and its two rows differ solely by whether that update is applied. At 256 frames the checkpoint scores 41.7 with the update against 18.3 without it, a paired difference of $+23.3$ points (38/10 discordant, exact McNemar $p = 6.2 \times 10^{-5}$), and $+15.8$ points over an update generated from a different video ($p = 0.003$). The effect does not survive dilution: $+7.5$ points at 1024 frames ($p = 0.14$) and -4.2 at 2048 ($p = 0.40$). Run on k=6 through k=16, the same contrast is null at every length.