SAEVIA: SPATIOTEMPORAL TOKEN COMPRESSION WITH ATTENTION COMPENSATION FOR EFFICIENT VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video large language models (VideoLLMs) have demonstrated strong capabilities in video understanding, yet processing numerous visual tokens incurs high inference latency and limits practical deployment. Existing token compression methods reduce temporal and spatial redundancy but often overlook compression-induced attention distortion, which can impair accuracy. We introduce Saevia, a training-free framework that combines spatiotemporal token compression with attention compensation. Specifically, position-wise temporal merging reduces temporal redundancy at each spatial position. Request-aware spatial pruning selects tokens by jointly considering visual salience and request relevance. During LLM prefill, sparsity-aware attention compensation mitigates the attention distortion caused by compression. Across three VideoLLMs and five video understanding benchmarks, Saevia preserves 95.3–99.2% of full-token average accuracy at 10% visual token retention. With dedicated Triton kernels and a FlashAttention-compatible attention compensation design, Saevia achieves 2.6–3.2× time-to-first-token (TTFT) speedups over full-token baselines on one NVIDIA A100 80GB GPU and up to 1.25× end-to-end throughput with 1K-token outputs.

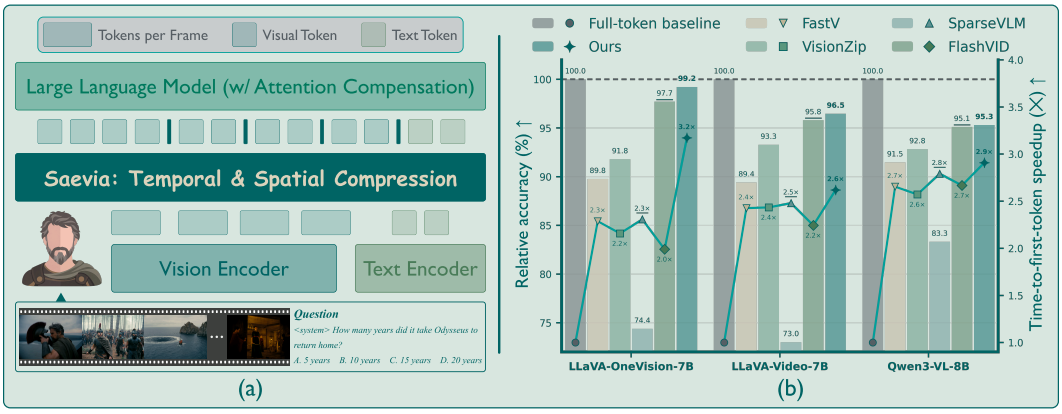

Figure 1: **Overview and performance of Saevia.** **(a)** We introduce Saevia, a training-free spatiotemporal token compression method with attention compensation for efficient video understanding. It uses temporal merging and spatial pruning after vision encoding, followed by attention compensation in LLM prefill to alleviate compression-induced distortion. **(b)** At 10% visual token retention, Saevia leads the five methods in five-benchmark average accuracy and time-to-first-token (TTFT) speedups (2.6–3.2×) across three models. Accuracy and speedups are relative to the corresponding model's full-token baseline. TTFT is measured on one real-world 720p video with one NVIDIA A100 80GB GPU. The **best** and second-best results are bold and underlined.

1 INTRODUCTION

Video large language models (VideoLLMs) (Li et al., 2025a; An et al., 2026; Zhang et al., 2024; Wang et al., 2024; Bai et al., 2025b;a; Zhu et al., 2025) have demonstrated extraordinary capabilities

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.

in video understanding tasks, but processing numerous visual tokens from videos incurs substantial inference overhead, hindering their deployment in production environments.

Training-free token compression offers a promising solution. It exploits the substantial redundancy in videos to reduce visual tokens while maintaining accuracy comparable to the full-token baseline. Requiring no additional training, it enables plug-and-play acceleration and rapid deployment. For example, FastV (Chen et al., 2024) uses early-layer LLM attention scores to prune less important visual tokens. Unlike FastV, VisionZip (Yang et al., 2025b) retains high-attention tokens and merges low-attention tokens into compact representations rather than discarding them. SparseVLM (Zhang et al., 2025d) uses relevant text tokens to assess visual token importance, guiding request-aware pruning and token recycling. FlashVID (Fan et al., 2026b) combines tree-based merging with attention and diversity-based token selection to jointly reduce temporal and spatial redundancy.

Despite these benefits, two limitations remain. First, many methods overlook attention distortion caused by compression-induced changes in the relative representation of visual tokens, potentially degrading accuracy. Second, many methods rely on attention scores, whose extraction may require switching from FlashAttention (Dao et al., 2022; Dao, 2024; Shah et al., 2024; Zadouri et al., 2026) to conventional attention implementations in the relevant layers. The cost of extracting attention scores, together with compression overhead, can limit end-to-end efficiency.

Based on these observations, we propose Saevia, a spatiotemporal token compression method with attention compensation for efficient video understanding. We first apply position-wise temporal merging to reduce temporal redundancy. Position-wise comparison captures fine-grained inter-frame redundancy, while merging similar tokens rather than dropping them helps preserve visual details. Request-aware spatial pruning then reduces spatial redundancy. It combines vision attention scores with visual–request relevance for frame-wise top-K selection, balancing token salience and request relevance to preserve both salient visual content and question-relevant information. Finally, sparsity-aware attention compensation alleviates attention distortion caused by compression-induced changes in relative token representation. We compute temporal and spatial token multiplicities separately and add their logarithms to the corresponding attention logit columns before softmax during LLM prefill. Temporal multiplicity records the number of original tokens represented by each retained token after merging, while spatial multiplicity measures the frame-wise ratio of vision attention mass before and after pruning. By combining the two multiplicities, we aim to recover the full-token attention distribution to mitigate the associated accuracy loss.

To improve inference efficiency, we optimize three aspects of the implementation. First, we retain the vision encoder's original FlashAttention implementation and reuse the queries and keys from its final layer to compute attention scores with a dedicated kernel. Second, we batch request-aware spatial pruning across frames and use a dedicated kernel to reduce repeated data processing and token gathering overhead. Finally, we implement attention compensation by appending 1 to each query vector and the scaled logarithm of token multiplicity to each key vector. This introduces compensation through the dot product, preserving compatibility with FlashAttention.

Putting these components and optimizations together, Saevia preserves 95.3–99.2% of the corresponding full-token average accuracy across five benchmarks at an aggressive 10% visual token retention ratio, while delivering 2.6–3.2× time-to-first-token (TTFT) speedups and up to 1.25× end-to-end throughput relative to full-token baselines with 1K-token outputs on one NVIDIA A100 80GB GPU. Extensive experiments demonstrate the effectiveness and efficiency of Saevia.

Figure 1 summarizes Saevia's design and performance. Our contributions are as follows:

- We introduce a training-free spatiotemporal token compression method combining position-wise temporal merging and request-aware spatial pruning. It reduces fine-grained redundancy while balancing token salience and request relevance to preserve useful visual information and improve video understanding accuracy.
- We propose sparsity-aware attention compensation to mitigate attention distortion caused by changes in the relative representation of visual tokens after compression. It aims to recover the full-token attention distribution and reduce the associated accuracy loss.
- Extensive experiments on three representative VideoLLMs and five video understanding benchmarks demonstrate substantial gains in inference efficiency while preserving high accuracy, supporting practical VideoLLM deployment.

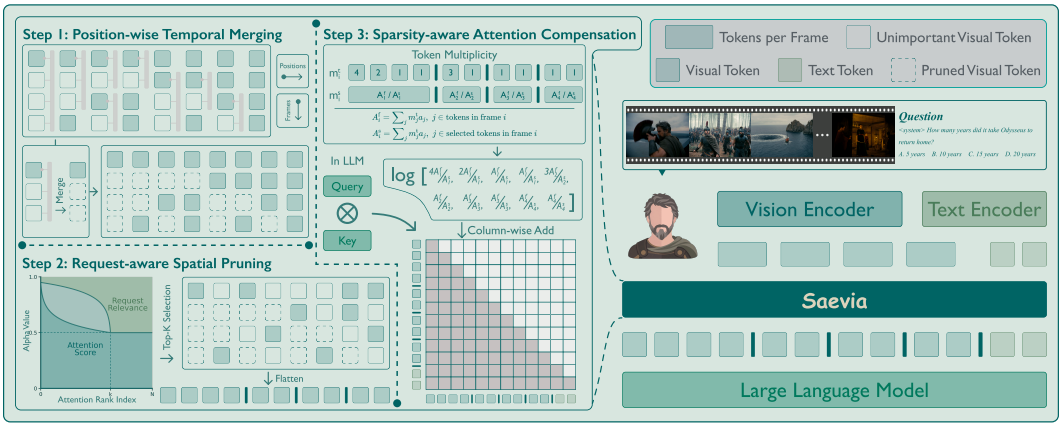

Figure 2: **Overview of our method. First**, temporal merging averages tokens at each spatial position across consecutive frames when adjacent-pair cosine similarities exceed the threshold, under maximum-span and merge-budget constraints. **Second**, spatial pruning combines vision attention scores with visual-request relevance for frame-wise top-K selection. **Finally**, attention compensation adds the logarithm of each retained visual token's overall multiplicity to its attention logit column before softmax during LLM prefill, alleviating compression-induced attention distortion.

2 METHOD

Figure 2 illustrates our method, which consists of temporal merging, spatial pruning, and attention compensation. We detail these components in Sections 2.2, 2.3, and 2.4, respectively.

2.1 PRELIMINARIES

Video LLM inference. We treat multiple-choice video understanding as a prefill-only workload because the first generated token usually determines the final answer and decoding averages no more than three steps. This workload has three stages: vision encoding, visual token processing, and LLM prefill. Vision encoding spans the vision encoder input through the projector output. Visual token processing covers operations between the projector output and the LLM input, including LLaVA's spatial downsampling. Many training-free methods, including ours, compress visual tokens here. LLM prefill processes the visual and text tokens to produce the first output token. Throughout this paper, visual token counts are measured at the LLM input, after any spatial downsampling.

Token multiplicity in attention. Training-free compression reduces visual token counts, but unequal regional compression rates change their relative representation. In a surveillance video, for example, merging background tokens into a few representative tokens reduces their share relative to the uncompressed foreground, distorting the original token distribution. To account for this distortion, we track each retained token's multiplicity and incorporate it into attention:

$$\mathbf{A} = \mathrm{softmax}\Big(\mathbf{Q}\mathbf{K}^{\mathsf{T}}/\sqrt{d} + \log \mathbf{m}\Big), \tag{1}$$

where $\mathbf{m}$ is a row vector of per-token multiplicities, with nonvisual entries set to 1. Adding $\log m_i$ to token i's attention logit gives it the same unnormalized attention weight as m_i identical tokens. This proportional-attention formulation was introduced by ToMe (Bolya et al., 2023). Inspired by this principle, we use separate temporal and spatial multiplicities to account for the distinct effects of merging and pruning on attention.

2.2 POSITION-WISE TEMPORAL MERGING

Reducing the substantial temporal redundancy in videos can improve inference efficiency. However, frame-level compression may overlook local details in highly redundant frames, leading to suboptimal accuracy. We therefore propose a fine-grained, position-wise strategy for temporal compression.

Compression unit identification. Given a video with N frames and P visual tokens per frame, we have visual embeddings $\mathbf{E}_v = [\mathbf{e}_{n,p}] \in \mathbb{R}^{N\times P\times d}$, where d is the hidden dimension. We first

Table 1: **Accuracy results for LLaVA-OneVision-7B.** 32 frames are uniformly sampled per video, yielding approximately 6K visual tokens. For each benchmark and retention ratio, the **best** and second-best results among all compression methods are shown in bold and underlined, respectively. For complete results at all five retention ratios, see Appendix E.1.

Method		Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
Full-token baseline		100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
FastV	ECCV'24	20%	53.10	56.48	55.89	52.06	57.79	55.06	95.60
VisionZip	CVPR'25	20%	54.75	58.07	57.74	53.33	60.26	56.83	98.67
SparseVLM	ICML'25	20%	53.75	55.78	56.45	48.17	59.04	54.64	94.86
FlashVID	ICLR'26 Oral	20%	54.65	**58.48**	57.94	**54.82**	61.50	57.48	99.80
Saevia	Ours	20%	**55.67**	58.15	**59.87**	54.00	**62.09**	**57.96**	**100.63**
FastV	ECCV'24	15%	52.90	54.89	54.84	51.38	56.63	54.13	93.98
VisionZip	CVPR'25	15%	53.53	55.96	57.13	52.66	58.79	55.61	96.56
SparseVLM	ICML'25	15%	51.75	52.74	53.81	47.05	56.20	52.31	90.82
FlashVID	ICLR'26 Oral	15%	54.65	58.04	57.88	**56.17**	60.67	57.48	99.80
Saevia	Ours	15%	**55.60**	**58.11**	**60.03**	54.53	**61.41**	**57.94**	**100.59**
FastV	ECCV'24	10%	51.88	52.30	53.35	46.82	54.16	51.70	89.77
VisionZip	CVPR'25	10%	51.85	52.74	55.10	47.57	57.11	52.87	91.80
SparseVLM	ICML'25	10%	41.88	44.26	39.67	42.63	45.76	42.84	74.38
FlashVID	ICLR'26 Oral	10%	53.73	**56.59**	57.36	53.93	59.78	56.28	97.71
Saevia	Ours	10%	**55.05**	56.37	**59.87**	**54.23**	**60.22**	**57.15**	**99.22**

compute the cosine similarities between tokens at the same spatial position in adjacent frames:

$$S_{n,p} = \cos(\mathbf{e}_{n,p}, \mathbf{e}_{n+1,p}), \qquad \mathbf{S} = [S_{n,p}] \in \mathbb{R}^{(N-1)\times P}. \tag{2}$$

For each spatial position, we place a boundary between adjacent tokens whose similarity is at or below the preset threshold. Together with the beginning and end of the video, these boundaries divide the token sequence at that spatial position into temporal compression units:

$$\mathcal{U}^p = \{\mathcal{U}^p_i \mid 1 \le i < |\mathcal{B}^p|\},\ \mathcal{U}^p_i = \left(\mathbf{e}_{B^p_{i-1}+1,p}, \ldots, \mathbf{e}_{B^p_i,p}\right),\ \mathcal{B}^p = \{0, N\} \cup \{n \mid S_{n,p} \le \tau\}, \tag{3}$$

where $\mathcal{U}^p$ is the set of compression units at spatial position p, $\mathcal{U}^p_i$ is the i-th unit, $\mathcal{B}^p$ is the corresponding boundary set, and τ is the similarity threshold. The boundaries are indexed in ascending order, beginning with $B^p_0 = 0$. Each unit contains tokens from one fixed spatial position across consecutive frames, while one spatial position may yield multiple units. By identifying redundancy at the token level, this strategy prevents local details from being overwhelmed by redundant frame-level content, increasing the density of informative content and supporting higher accuracy.

Position-wise token merging. For each temporal compression unit, we replace its first token with the unit's average embedding and discard all remaining tokens:

$$\mathcal{U}^p_i[0] \leftarrow \textstyle\sum_{\mathbf{e}\in\mathcal{U}^p_i} \mathbf{e} \ / \ |\mathcal{U}^p_i|, \qquad \mathcal{U}^p_i[1:] \leftarrow \emptyset. \tag{4}$$

For highly similar and redundant tokens, merging typically preserves more information than directly dropping them, reducing redundancy while retaining useful content.

Maximum-span constraint. Even when similarities between adjacent tokens exceed the threshold, merging a long temporal unit into one token can severely distort the video's temporal distribution and harm accuracy. We therefore limit each compression unit to at most $s_{\max}$ original tokens. We split any unit $\mathcal{U}^p_i$ with $|\mathcal{U}^p_i| > s_{\max}$ into c contiguous units whose sizes differ by at most one, then reindex all subsequent units in their original order:

$$\mathcal{U}^p_i \to (\mathcal{U}^p_i, \ldots, \mathcal{U}^p_{i+c-1}), \quad \mathcal{U}^p_k \to \mathcal{U}^p_{k+c-1} \quad \text{for } k > i, \quad c = \left\lceil |\mathcal{U}^p_i| \,/\, s_{\max} \right\rceil. \tag{5}$$

This limit avoids extreme merging and better preserves the original video's temporal distribution.

Merge-budget constraint. Since the number of eligible merge connections depends on video content, unconstrained temporal merging may leave fewer tokens than the preset retention budget. After applying the similarity and maximum-span constraints, let $\mathcal{B}^p$ contain all boundaries at position p. Then $M = \sum_{p=1}^{P}(|\mathcal{B}^p| - 1)$ is the total number of tokens that would remain after merging. Given the target retention ratio r and the corresponding target token count $K = \mathrm{round}(rNP)$, we preserve

Table 2: **Accuracy results for LLaVA-Video-7B and Qwen3-VL-8B-Instruct.** The two models use 64 and 128 uniformly sampled frames per video, respectively. For each model, benchmark, and retention ratio, the **best** and second-best results among all compression methods are shown in bold and underlined, respectively. For complete results at all five retention ratios, see Appendix E.1.

Method		Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-Video-7B									
Full-token baseline		100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
FastV	ECCV'24	20%	54.63	59.19	53.57	53.10	59.15	55.93	93.29
VisionZip	CVPR'25	20%	**56.45**	62.00	56.03	55.87	63.12	58.69	97.90
SparseVLM	ICML'25	20%	54.15	59.44	52.65	52.13	58.99	55.47	92.53
FlashVID	ICLR'26 Oral	20%	55.90	**62.11**	55.77	**56.77**	63.38	58.79	98.07
Saevia	Ours	20%	55.93	61.89	**57.13**	56.17	**65.31**	**59.29**	**98.90**
FastV	ECCV'24	10%	53.20	55.89	50.31	50.79	57.80	53.60	89.41
VisionZip	CVPR'25	10%	54.75	58.78	52.83	52.66	60.56	55.92	93.28
SparseVLM	ICML'25	10%	43.87	46.63	35.84	43.83	48.70	43.77	73.01
FlashVID	ICLR'26 Oral	10%	**55.77**	60.19	54.60	54.23	62.39	57.44	95.81
Saevia	Ours	10%	54.65	**60.93**	**55.85**	**54.90**	**62.85**	**57.84**	**96.48**
Qwen3-VL-8B-Instruct									
Full-token baseline		100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
FastV	ECCV'24	20%	**66.20**	66.11	71.18	62.00	65.39	66.18	95.87
VisionZip	CVPR'25	20%	65.35	**67.59**	**71.91**	63.05	68.27	67.23	97.39
SparseVLM	ICML'25	20%	64.98	66.96	71.02	61.41	67.37	66.35	96.11
FlashVID	ICLR'26 Oral	20%	64.52	66.59	71.18	63.50	68.20	66.80	96.76
Saevia	Ours	20%	64.85	67.22	71.40	**63.95**	**68.93**	**67.27**	**97.45**
FastV	ECCV'24	10%	63.88	63.78	69.51	58.56	60.01	63.15	91.48
VisionZip	CVPR'25	10%	61.30	64.00	69.81	59.99	65.30	64.08	92.83
SparseVLM	ICML'25	10%	56.17	59.93	65.59	52.28	53.63	57.52	83.32
FlashVID	ICLR'26 Oral	10%	62.90	65.48	70.18	**62.98**	**66.75**	65.66	95.11
Saevia	Ours	10%	**64.30**	**66.15**	**70.68**	61.33	66.48	**65.79**	**95.31**

these boundaries if $M \geq K$. Otherwise, we add $K - M$ boundaries at the remaining connections with the lowest cosine similarities across all spatial positions to cap the number of removed tokens:

$$\mathcal{B}^p \leftarrow \mathcal{B}^p \cup \{n \mid (n,p) \in \mathrm{BottomK}(\mathbf{S}, K-M)\}, \qquad \text{if } M < K, \tag{6}$$

where $\mathrm{BottomK}(\mathbf{S}, K)$ selects the K lowest-similarity connections in $\mathbf{S}$ that are not cut by the existing boundaries and returns their indices (n, p). Each additional boundary increases the retained token count by one. The added boundaries therefore bring the count to K without changing any existing boundaries. This allows content-adaptive merging within the preset compression budget.

2.3 Request-Aware Spatial Pruning

Temporal merging removes redundancy across frames based solely on video content, so the number of tokens it removes varies across videos. Spatial pruning complements this stage by selecting informative tokens within each frame to reach the preset overall retention ratio.

Visual-request relevance computation. After projection and spatial downsampling (if any), we compute each visual token's raw request-relevance score by averaging its cosine similarities with all tokens in the question and answer choices, excluding the system prompt. We then apply softmax to the raw scores within each frame. To account for the reduction in attention mass caused by spatial downsampling, we multiply the normalized relevance scores by the frame's total attention score. This places relevance and attention scores on the same scale. The process is expressed as follows:

$$q^{\mathrm{raw}}_{n,p} = \frac{1}{L}\sum_{\ell=1}^{L} \cos(\mathbf{e}_{n,p}, \mathbf{e}^{\mathrm{r}}_{\ell}), \qquad \mathbf{q}_n = \mathrm{softmax}(\mathbf{q}^{\mathrm{raw}}_n)\sum_{p=1}^{P} a_{n,p}, \tag{7}$$

where L is the number of request tokens, $\mathbf{e}^{\mathrm{r}}_{\ell}$ is the embedding of request token ℓ, $\mathbf{q}^{\mathrm{raw}}_n = (q^{\mathrm{raw}}_{n,1}, \ldots, q^{\mathrm{raw}}_{n,P})$ collects the raw relevance scores in frame n, and $a_{n,p}$ is the attention score of token p in frame n. The attention and normalized relevance scores undergo the same temporal merging as the visual tokens: both are averaged within each unit. Request relevance reflects how closely visual tokens relate to the request, helping retain visual information for answering the question.

Table 3: **Efficiency results for three models.** Efficiency is measured on a single real-world 720p video with one NVIDIA A100 80GB GPU. Accuracy is averaged across five benchmarks. For each model and metric, the **best** and second-best results among all compression methods are shown in bold and underlined, respectively. For results at more retention ratios, see Appendix E.3.

Method		Ret. Ratio	TTFT (ms) ↓				Throughput (tokens/s) ↑		Accuracy	
			Overall	Vision Enc.	Token Proc.	LLM Prefill	Prefill	End-to-End	Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	10%	277.23 (2.29×)	117.41	62.71	96.97	22,960 (2.29×)	40.20 (1.03×)	51.70	89.77
VisionZip	CVPR'25	10%	293.91 (2.16×)	124.95	108.61	**60.23**	21,657 (2.16×)	40.02 (1.03×)	52.87	91.80
SparseVLM	ICML'25	10%	274.90 (2.31×)	116.77	63.00	95.00	23,154 (2.31×)	40.00 (1.02×)	42.84	74.38
FlashVID	ICLR'26 Oral	10%	318.65 (1.99×)	124.58	120.81	73.14	19,975 (1.99×)	**40.34** (1.03×)	56.28	97.71
Saevia	Ours	10%	**199.95** (3.17×)	**116.35**	**21.55**	61.94	**31,833** (3.17×)	40.11 (1.03×)	**57.15**	**99.22**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	10%	498.62 (2.43×)	340.37	4.66	153.42	22,029 (2.43×)	40.06 (1.02×)	53.60	89.41
VisionZip	CVPR'25	10%	497.09 (2.43×)	353.30	53.75	**89.87**	22,097 (2.43×)	40.12 (1.02×)	55.92	93.28
SparseVLM	ICML'25	10%	488.00 (2.48×)	340.13	**4.61**	143.08	22,508 (2.48×)	**41.27** (1.05×)	43.77	73.01
FlashVID	ICLR'26 Oral	10%	539.33 (2.24×)	350.34	85.53	103.28	20,367 (2.24×)	40.72 (1.03×)	57.44	95.81
Saevia	Ours	10%	**461.97** (2.62×)	**338.88**	32.24	90.69	**23,777** (2.62×)	40.80 (1.03×)	**57.84**	**96.48**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	10%	2052.80 (2.65×)	**1429.02**	27.54	596.05	15,791 (2.65×)	25.51 (1.23×)	63.15	91.48
VisionZip	CVPR'25	10%	2118.89 (2.57×)	1599.48	231.41	**287.82**	15,298 (2.57×)	25.80 (1.25×)	64.08	92.83
SparseVLM	ICML'25	10%	1951.85 (2.79×)	1438.38	**23.82**	489.45	16,607 (2.79×)	25.29 (1.22×)	57.52	83.32
FlashVID	ICLR'26 Oral	10%	2042.98 (2.67×)	1610.18	120.78	311.84	15,867 (2.67×)	25.23 (1.22×)	65.66	95.11
Saevia	Ours	10%	**1873.54** (2.91×)	1450.93	122.14	300.32	**17,302** (2.91×)	**25.91** (1.25×)	**65.79**	**95.31**

Request-aware re-ranking. We incorporate request relevance into attention-based ranking. Within each frame n, we denote the candidate index set by $\mathcal{C}_n$ and rank candidates by descending attention score. Their indices i are zero-based and remain fixed during re-ranking. The attention weight α_i decreases logarithmically over the top-K_n candidates in this original ranking and remains fixed thereafter. We vary the schedule across models to balance visual salience and request relevance:

$$\alpha_i = \begin{cases} \textit{concave:}\ \alpha_{\text{tail}} + (\alpha_{\text{head}} - \alpha_{\text{tail}}) \log(K_n - i + 1) / \log(K_n + 1), & \\ \textit{convex:}\quad \alpha_{\text{head}} - (\alpha_{\text{head}} - \alpha_{\text{tail}}) \log(i + 1) / \log(K_n + 1), & \end{cases} \quad \begin{matrix} \alpha_i = \alpha_{\text{tail}} \\ \text{for } i \geq K_n. \end{matrix} \tag{8}$$

Here, K_n is the number of tokens to retain in frame n, and $0 \leq \alpha_{\text{tail}} \leq \alpha_{\text{head}} \leq 1$. The concave and convex branches apply to $0 \leq i < K_n$. This schedule assigns greater attention weight to highly ranked candidates while gradually increasing the influence of request relevance toward the selection boundary. Beyond the top-K_n range, the attention weight remains at α_{tail}. We then combine the attention and request-relevance scores to re-rank the candidates for top-K_n selection:

$$\mathcal{C}_n^\star = \text{TopK}((\text{score}_i)_{i \in \mathcal{C}_n}, K_n), \qquad \text{score}_i = \alpha_i a_i + (1 - \alpha_i) q_i, \tag{9}$$

where TopK returns the indices of the K_n highest-scoring candidates, forming the retained set $\mathcal{C}_n^\star \subseteq \mathcal{C}_n$, while a_i and q_i denote the attention and request-relevance scores of candidate i, respectively.

Frame-wise top-K selection. After temporal merging, we distribute the final token budget across frames according to their remaining candidate counts. Whenever the budget permits, each nonempty frame retains at least one token. We then select each frame's top-K_n candidates using the combined score. Although attention-based top-K selection provides good baseline accuracy, request relevance helps preserve visual tokens more useful for answering the question. The logarithmic decay schedule prioritizes highly salient tokens, balancing visual salience with request relevance across models.

2.4 Sparsity-Aware Attention Compensation

Temporal merging and spatial pruning reach the preset retention ratio but distort the full-token attention distribution. After temporal merging, one retained token may represent multiple original tokens but still contributes to attention as only one token. Spatial pruning further removes the attention contributions of discarded tokens. We therefore propose sparsity-aware attention compensation to better

Table 4: **Component ablation study across five benchmarks.** Spatial pruning is replaced by *attention-based top-K* or *random* selection under the same token budget. For each model and metric, the **best** and second-best variants are bold and underlined. For detailed results, see Appendix E.4.

Method	Ret. Ratio	*LLaVA-OneVision-7B*		*LLaVA-Video-7B*		*Qwen3-VL-8B-Instruct*	
		Avg.	Pct. (%)	Avg.	Pct. (%)	Avg.	Pct. (%)
Full-token baseline	100%	57.60	100.00	59.95	100.00	69.03	100.00
w/o Temporal Merging	15%	56.29	97.73	58.13	96.96	66.25	95.97
w/o Spatial Pruning *(Attn Top-K)*	15%	57.65	100.09	58.73	97.96	66.80	96.77
w/o Spatial Pruning *(Random)*	15%	56.90	98.78	57.56	96.01	65.72	95.20
w/o Attention Compensation	15%	57.38	99.62	58.23	97.13	66.71	96.64
Saevia	15%	**57.94**	**100.59**	**58.82**	**98.12**	**66.90**	**96.91**

recover the full-token attention distribution. During LLM prefill, it adds a bias to the corresponding visual token columns of the attention logit matrix before softmax. In this subsection, i indexes the original flattened visual token sequence, with $\mathcal{C}_n$, $\mathcal{C}_n^\star$, and the associated scores reindexed to match.

Compensation for temporal merging. For each visual token, we define its temporal token multiplicity m_i^{t} based on whether it is merged and whether it is the first token in its compression unit:

$$m_i^{\mathrm{t}} = |\mathcal{U}_j^p| \text{ } \textit{(merged, first)}, \quad m_i^{\mathrm{t}} = \text{None } \textit{(merged, non-first)}, \quad m_i^{\mathrm{t}} = 1 \text{ } \textit{(unmerged)}, \tag{10}$$

where $\mathcal{U}_j^p$ is the compression unit containing token i before merging, and None marks discarded tokens. Defining temporal multiplicity by the number of represented tokens provides a straightforward and natural correction for the attention distortion introduced by temporal merging.

Compensation for spatial pruning. For each visual token retained after spatial pruning, we define its spatial token multiplicity m_i^{s} as the total attention mass of all candidates in the corresponding frame divided by the attention mass of the selected candidates:

$$m_i^{\mathrm{s}} = \textstyle\sum_{j\in\mathcal{C}_n} m_j^{\mathrm{t}} a_j \;/\; \sum_{j\in\mathcal{C}_n^\star} m_j^{\mathrm{t}} a_j, \qquad i \in \mathcal{C}_n^\star, \tag{11}$$

where $m_j^{\mathrm{t}} a_j$ is the attention mass of the original tokens represented by candidate j. All retained candidates in frame n share the same spatial multiplicity. Within each temporal compression unit, the same vision attention scores are summed to obtain attention mass for compensation, whereas they are averaged for request-aware re-ranking, as described above. Since spatial pruning discards tokens rather than merging them, using attention mass instead of token count accounts for the importance of the discarded tokens, correcting the loss of attention mass while avoiding overcompensation.

Joint compensation. During LLM prefill, we combine temporal and spatial multiplicities by adding their logarithms to the corresponding token columns of the attention logit matrix before softmax:

$$\mathbf{L}'[:,i] = \mathbf{L}[:,i] + \log m_i^{\mathrm{t}} + \log m_i^{\mathrm{s}}, \qquad \mathbf{A} = \mathrm{softmax}(\mathbf{L}'), \tag{12}$$

where $\mathbf{L}$ and $\mathbf{L}'$ are the attention logit matrices before and after compensation, respectively, and $\mathbf{L}[:,i]$ denotes the column corresponding to retained visual token i. We update all visual token columns before applying row-wise softmax. Nonvisual token columns remain unchanged, equivalent to setting both multiplicities to 1. The two multiplicities account for the distinct effects of temporal merging and spatial pruning, recovering the original full-token attention distribution as much as possible while avoiding overcompensation and preserving accuracy.

3 Experiments

3.1 Experimental Settings

Models. We evaluate our method on three representative VideoLLMs: LLaVA-OneVision-7B (Li et al., 2025a), LLaVA-Video-7B (Zhang et al., 2024), and Qwen3-VL-8B-Instruct (Bai et al., 2025a). For accuracy evaluation, we uniformly sample 32 and 64 frames per video for LLaVA-OneVision and LLaVA-Video, yielding approximately 6K and 11K visual tokens, respectively. Qwen3-VL uses 128 uniformly sampled frames per video, with a resolution-dependent visual token count.

Benchmarks. We evaluate our method on five video understanding benchmarks: MVBench (Li et al., 2024a), Video-MME (Fu et al., 2025a), EgoSchema (Mangalam et al., 2023),

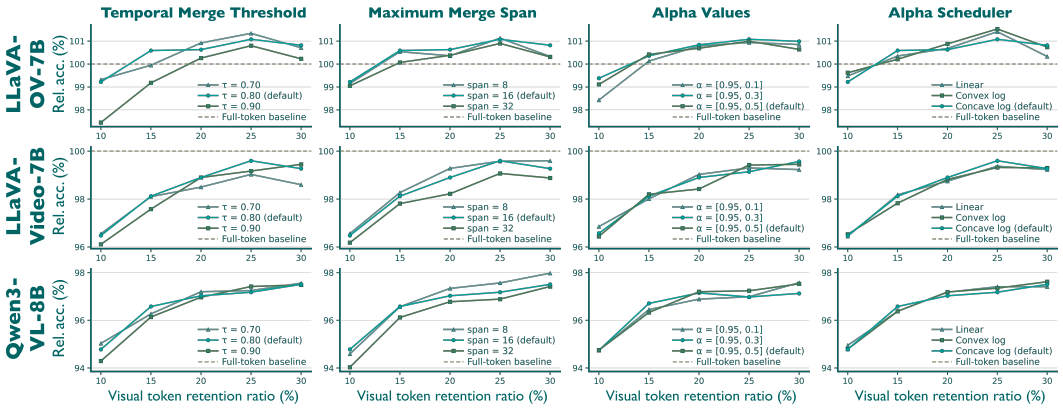

Figure 3: **Hyperparameter sensitivity of Saevia.** We vary four hyperparameters across three models and five benchmarks. All experiments use C1 as the base configuration, varying one hyperparameter at a time. For detailed numerical results, see Appendix E.5.

LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2025). We use Video-MME without subtitles, the full EgoSchema set, and the LongVideoBench validation set. For MLVU, we report M-Avg on the development set. Further benchmark details are provided in Appendix C.

Baselines. We compare Saevia with four training-free baselines: FastV (Chen et al., 2024) prunes visual tokens using early-layer LLM attention scores. VisionZip (Yang et al., 2025b) retains tokens with high vision attention scores and merges the rest into compact contextual representations. SparseVLM (Zhang et al., 2025d) uses relevant text tokens to guide visual token pruning and recycles selected pruned tokens to preserve information. FlashVID (Fan et al., 2026b) combines attention and diversity-based token selection with tree-based merging to reduce spatiotemporal redundancy.

Hyperparameters. For LLaVA-OneVision and LLaVA-Video, we use compound C1: $\{\tau = 0.8, s_{\max} = 16, (\alpha_{\text{head}}, \alpha_{\text{tail}}) = (0.95, 0.5), \alpha \text{ scheduler} = \text{concave log}\}$. For Qwen3-VL, we use compound C2: $\{\tau = 0.7, s_{\max} = 8, (\alpha_{\text{head}}, \alpha_{\text{tail}}) = (0.95, 0.5), \alpha \text{ scheduler} = \text{convex log}\}$.

Implementation. We use Triton (Tillet et al., 2019) kernels to accelerate vision attention score recomputation and request-aware spatial pruning. Attention compensation is compatible with FlashAttention. For a fair comparison, we implement and evaluate all baselines in the same framework. For algorithm and baseline deployment details, see Appendix B and Appendix D.1, respectively.

3.2 Main Results

Accuracy results. Table 1 shows that Saevia leads the compared methods in five-benchmark average accuracy at all three retention ratios on LLaVA-OneVision. It preserves 99.2% of full-token average accuracy at 10% visual token retention and even exceeds it at higher ratios. Table 2 shows that Saevia also leads in average accuracy on LLaVA-Video and Qwen3-VL at 10% retention, with less than 5% relative loss against their full-token baselines. These results show generalizability and adaptability across backbones. See Appendices E.1 and E.2 for more results and video-duration breakdowns.

Efficiency results. Table 3 reports inference efficiency on a single real-world 720p video with one NVIDIA A100 80GB GPU. At 10% retention, Saevia achieves 2.6–3.2× TTFT speedups over the full-token baselines, the highest among the compared methods. Meanwhile, it improves end-to-end throughput by a factor of 1.03–1.25 compared with the corresponding full-token baselines when generating 1K tokens. Saevia achieves these efficiency gains while maintaining the highest accuracy among the compared methods. See Appendix E.3 for additional results.

3.3 Ablation Studies

Component ablations. Table 4 examines the contribution of each component. We replace request-aware spatial pruning with attention-based top-K or random selection to maintain the same token budget. Across all three models, removing any component lowers the final average accuracy. These results support the effectiveness and soundness of Saevia's design.

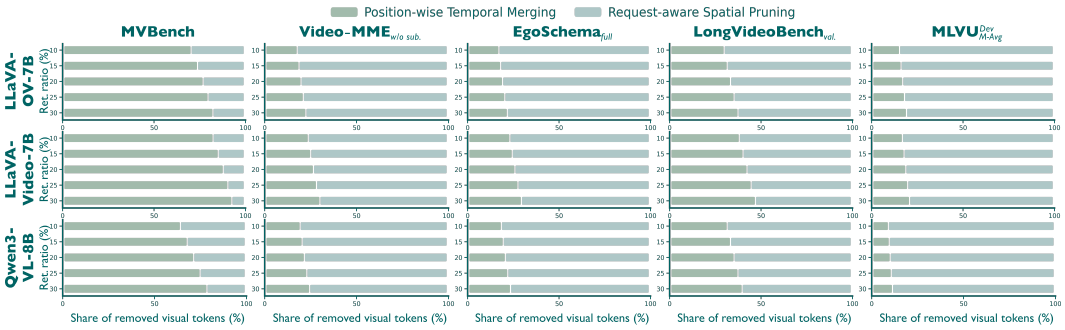

Figure 4: **Visual token reduction breakdown.** Shares (%) of all removed visual tokens attributable to temporal merging and spatial pruning, across three models and five benchmarks.

Hyperparameter sensitivity. Figure 3 evaluates sensitivity to four hyperparameters across three models. Starting from C1, we vary one hyperparameter at a time. For each model and retention ratio, the five-benchmark average accuracy varies by at most 1.08 percentage points within a sweep, indicating limited sensitivity within the tested ranges. Detailed results are provided in Appendix E.5.

3.4 Analysis

Compression breakdown. Figure 4 shows each stage's contribution. With the model, video, and hyperparameters fixed, temporal merging removes a fixed number of tokens unless capped by the merge budget. Higher retention ratios reduce spatial pruning, increasing temporal merging's share of total reduction. Temporal share is highest on MVBench and generally lowest for Qwen3-VL. These results show how spatial pruning complements content-dependent temporal merging to meet the token budget and highlight different redundancy patterns across models and benchmarks.

4 Related Work

See Appendix A for the full discussion.

Spatiotemporal token compression. FrameFusion (Fu et al., 2025b) merges spatially aligned tokens and prunes by importance; FlashVID (Fan et al., 2026b) combines attention and diversity-based selection with tree-based merging and late LLM pruning. Saevia completes compression before LLM prefill and bounds position-wise temporal unit spans.

Text-guided token compression. Text-guided methods use LLM attention (Zhuang et al., 2025) or visual–text relevance (Zhang et al., 2025c; Song et al., 2025). AdaFV (Han et al., 2025) splits a pre-LLM budget between visual salience and text relevance rankings; Saevia weights both signals by the original vision attention rank to score tokens for frame-wise selection.

Attention compensation. ToMe (Bolya et al., 2023) adds a logarithmic bias based on merged-token size; RESTORE (Cho et al., 2026) also considers positional distance. Saevia uses represented-token counts for temporal compensation and per-frame ratios of total to retained attention mass for spatial compensation, avoiding count-based overcompensation while supporting FlashAttention.

5 Conclusion

We present Saevia, a training-free framework for efficient video understanding via spatiotemporal token compression and attention compensation. Position-wise temporal merging first merges similar tokens to reduce inter-frame redundancy. Then request-aware spatial pruning prunes tokens by visual salience and request relevance to reduce intra-frame redundancy. Finally, during LLM prefill, sparsity-aware attention compensation mitigates compression-induced distortion due to changes in relative token representation. At 10% visual token retention, Saevia retains 95.3–99.2% of full-token average accuracy across three VideoLLMs and five benchmarks, with 2.6–3.2× TTFT speedups and 1.03–1.25× end-to-end throughput relative to full-token baselines with 1K-token outputs on one NVIDIA A100 GPU. As a training-free method compatible with FlashAttention, Saevia can serve as a plug-and-play solution for efficient VideoLLM inference, supporting production deployment.

AI USE STATEMENT

We used AI tools to support our review of related research. During method development, the authors designed the core software architecture, while AI tools assisted with code development for deployment. The manuscript was written by the authors, with AI tools used to polish the language and correct grammatical and spelling errors.

REPRODUCIBILITY STATEMENT

We aim to make our implementation and evaluation transparent and reproducible. Section 2 presents Saevia, and Appendix B provides pseudocode and implementation details for all three components, including budget allocation, token ordering, and attention compensation. Section 3.1 specifies the models, frame sampling, and hyperparameters, while Appendix C describes the benchmark subsets and evaluation metrics. For detailed implementation settings, see Appendix D, which documents baseline and Saevia deployment, token budget conventions, numerical precision, random seeds, and evaluation prompts. Appendix E.3 specifies the efficiency profiling hardware, workloads, warm-up and repeated runs, and metric definitions. Appendix E provides complete accuracy and efficiency results across retention ratios, component ablations, and hyperparameter sweeps.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie 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 *NeurIPS*, 2022. URL https://arxiv.org/abs/2204.14198.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, 2025. URL https://arxiv.org/abs/2503.02175.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026. doi: 10.48550/arXiv.2605.25979. URL https://arxiv.org/abs/2605.25979.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. HiRED: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *AAAI*, 2025. URL https://arxiv.org/abs/2408.10945.

Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. AgilePruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. In *ICLR*, 2026. URL https://arxiv.org/abs/2603.01236.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 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, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. MADTP: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *CVPR*, 2024. URL https://arxiv.org/abs/2403.02991.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *ACL*, 2023. URL https://aclanthology.org/2023.acl-long.721/.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal LLM. In *CVPR*, 2024. URL https://arxiv.org/abs/2312.06742.

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 *ECCV*, 2024. URL https://arxiv.org/abs/2403.06764.

Xueyi Chen, Keda Tao, Kele Shao, and Huan Wang. StreamingTOM: Streaming token compression for efficient video understanding. In *CVPR*, pp. 24675–24685, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_StreamingTOM_Streaming_Token_Compression_for_Efficient_Video_Understanding_CVPR_2026_paper.html.

Hyeonwoo Cho, Donghyeon Baek, Yewon Kim, and Bumsub Ham. Improving visual token reduction via rectifying distortions for efficient multimodal LLM inference. In *ICML*, 2026. URL https://arxiv.org/abs/2606.01711.

Joonmyung Choi, Sanghyeok Lee, Byungoh Ko, Eunseo Kim, Jihyung Kil, and Hyunwoo J. Kim. Representation shift: Unifying token compression with FlashAttention. In *ICCV*, 2025. URL https://arxiv.org/abs/2508.00367.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2305.06500.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-Awareness. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2205.14135.

Rong Fan, Kaiyan Xiao, Minghao Zhu, Liuyi Wang, Kai Dai, and Zhao Yang. GroundVTS: Visual token sampling in multimodal large language models for video temporal grounding. In *CVPR*, 2026a. URL https://arxiv.org/abs/2604.02093.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. FlashVID: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *ICLR*, 2026b. URL https://openreview.net/forum?id=H6rDX4w6Al.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *CVPR*, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining similarity and importance for video token reduction on large vision language models. In *ICCV*, pp. 22654–22663, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/html/Fu_FrameFusion_Combining_Similarity_and_Importance_for_Video_Token_Reduction_on_ICCV_2025_paper.html.

Jiayi Han, Liang Du, Yiwen Wu, Xiangguo Zhou, Hongwei Du, and Weibo Zheng. AdaFV: Rethinking of visual-language alignment for VLM acceleration. *arXiv preprint arXiv:2501.09532*, 2025. URL https://arxiv.org/abs/2501.09532.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. ZipVL: Accelerating vision-language models through dynamic token sparsity. In *ICCV*, 2025. URL https://arxiv.org/abs/2410.08584.

Lianyu Hu, Liqing Gao, Fanhua Shang, Liang Wan, and Wei Feng. iLLaVA: An image is worth fewer than 1/3 input tokens in large multimodal models. In *ICLR*, 2026. URL https://arxiv.org/abs/2412.06263.

Mouxiao Huang, Borui Jiang, Dehua Zheng, Hailin Hu, Kai Han, and Xinghao Chen. PPE: Positional preservation embedding for token compression in multimodal large language models. In *ICLR*, 2026a. URL https://arxiv.org/abs/2510.22936.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of ACL*, 2025. URL https://aclanthology.org/2025.findings-acl.1024/.

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong Yu, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by VLM token pruning. *arXiv preprint arXiv:2602.02951*, 2026b. URL https://arxiv.org/abs/2602.02951.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video LLMs. In *ICCV*, 2025. URL https://arxiv.org/abs/2507.07990.

Jing Jiang, Yiran Ling, Ruonan Li, Dimitrios Stamoulis, and Jie Liu. CoFiE: Coarse-to-fine evidence selection for efficient streaming video understanding. *arXiv preprint arXiv:2609.03675*, 2026. URL https://arxiv.org/abs/2609.03675.

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. *TMLR*, 2025a.

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 *ICML*, 2023. URL https://arxiv.org/abs/2301.12597.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal LLM. *IJCV*, 2025b. doi: 10.1007/s11263-025-02491-7. URL https://doi.org/10.1007/s11263-025-02491-7.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. VideoChat-Flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024b. URL https://arxiv.org/abs/2501.00574.

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *ECCV*, 2024c. URL https://arxiv.org/abs/2311.17043.

Haiji Liang, Pengfei Zhou, Zhenglin Wan, Wei Wang, Yang You, and Wangbo Zhao. Beyond one-size-fits-all: Sample-adaptive strategy routing for vision token pruning in MLLMs. *arXiv preprint arXiv:2609.10346*, 2026. URL https://arxiv.org/abs/2609.10346.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. URL https://arxiv.org/abs/2202.07800.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024. URL https://arxiv.org/abs/2411.10803.

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. In *CVPR*, 2025. URL https://arxiv.org/abs/2412.04468.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. ApET: Approximation-error guided token compression for efficient VLMs. In *CVPR*, 2026. URL https://arxiv.org/abs/2602.19870.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023.

Minyoung Park, Taehun Kong, and Sangjun Ahn. DynaTok: Temporally adaptive and positional bias-aware token compression for Video-LLMs. *arXiv preprint arXiv:2605.19322*, 2026. URL https://arxiv.org/abs/2605.19322.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. URL https://arxiv.org/abs/2106.02034.

Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *NeurIPS*, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 68658–68685. Curran Associates, Inc., 2024. doi: 10.52202/079017-2193. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7ede97c3e082c6df10a8d6103a2eebd2-Paper-Conference.pdf.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025. URL https://arxiv.org/abs/2403.15388.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. In *NeurIPS*, 2025. doi: 10.52202/085713-4525. URL https://papers.nips.cc/paper_files/paper/2025/hash/c573258c38d0a3919d8c1364053c45df-Abstract-Conference.html.

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. *TMLR*, 2026. URL https://openreview.net/forum?id=G2od9JVHkE.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *NeurIPS*, 2025a. URL https://arxiv.org/abs/2503.11187.

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 *ICML*, 2025b. URL https://proceedings.mlr.press/v267/shen25j.html.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X. Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal LLMs. In *COLING*, pp. 7614–7623, 2025. URL https://aclanthology.org/2025.coling-main.508/.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. LVPruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of NAACL*, 2025. URL https://aclanthology.org/2025.findings-naacl.242/.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *CVPR*, pp. 18992–19001, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Tao_DyCoke_Dynamic_Compression_of_Tokens_for_Fast_Video_Large_Language_CVPR_2025_paper.html.

Philippe Tillet, H. T. Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19. ACM, 2019. doi: 10.1145/3315508.3329973. URL https://doi.org/10.1145/3315508.3329973.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. FastVLM: Efficient vision encoding for vision language models. In *CVPR*, 2025. URL https://arxiv.org/abs/2412.13303.

Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, Shuai Xiao, and Enzo Tartaglione. FOLDER: Accelerating multi-modal large language models with enhanced performance. In *ICCV*, 2025. URL https://arxiv.org/abs/2501.02430.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *CVPR*, pp. 40559–40568, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_EarlyTom_Early_Token_Compression_Completes_Fast_Video_Understanding_CVPR_2026_paper.html.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. URL https://arxiv.org/abs/2409.12191.

Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. In *CVPR*, 2026b. URL https://arxiv.org/abs/2512.00891.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *EMNLP*, 2025. doi: 10.18653/v1/2025.emnlp-main.505. URL https://aclanthology.org/2025.emnlp-main.505/.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS*, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *CVPR*, 2025. URL https://arxiv.org/abs/2410.17247.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, and Bo Yuan. TopV: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *CVPR*, 2025a. URL https://arxiv.org/abs/2503.18278.

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. In *CVPR*, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.pdf.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. DeCo: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*, 2024. URL https://arxiv.org/abs/2405.20985.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. TimeChat-Online: 80% visual tokens are naturally redundant in streaming videos. In *ACMMM*, 2025. URL https://arxiv.org/abs/2504.17343.

Linli Yao, Long Xing, Yang Shi, Sida Li, Yuanxin Liu, Yuhao Dong, Yi-Fan Zhang, Lei Li, Qingxiu Dong, Xiaoyi Dong, Qidong Huang, Haotian Wang, Feng Wu, Yuanxing Zhang, Pengfei Wan, Zhouchen Lin, and Xu Sun. Towards efficient multimodal large language models: A survey on token compression. TechRxiv, January 2026. URL https://doi.org/10.36227/techrxiv.176823010.07236701/v1.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *CVPR*, 2025a. URL https://arxiv.org/abs/2412.00447.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. VoCo-LLaMA: Towards vision compression with large language models. In *CVPR*, 2025b. URL https://arxiv.org/abs/2406.12275.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. VisionTrim: Unified vision token compression for training-free MLLM acceleration. In *ICLR*, 2026. URL https://arxiv.org/abs/2601.22674.

Ted Zadouri, Markus Hoehnerbach, Jay Shah, Vijay Thakkar, and Tri Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. In A. Chowdhery and Z. Jia (eds.), *Proceedings of Machine Learning and Systems*, volume 8, pp. 912–926. MLSys, 2026. URL https://proceedings.mlsys.org/paper_files/paper/2026/file/ae8b0b5838ba510daff1198474e7b984-Paper-Conference.pdf.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. In *ICCV*, 2025a. URL https://arxiv.org/abs/2412.01818.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *NeurIPS*, 2025b. URL https://papers.nips.cc/paper_files/paper/2025/hash/2433fec2144ccf5fea1c9c5ebdbc3924-Abstract-Conference.html.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient image and video large multimodal models with one vision token. In *ICLR*, 2025c. URL https://arxiv.org/abs/2501.03895.

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 *ICML*, pp. 74840–74857, 2025d. URL https://proceedings.mlr.press/v267/zhang25s.html.

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

Xiangtian Zheng, Zishuo Wang, and Yuxin Peng. TiFRe: Text-guided video frame reduction for efficient video multi-modal large language models. *arXiv preprint arXiv:2602.08861*, 2026. URL https://arxiv.org/abs/2602.08861.

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 *CVPR*, pp. 13691–13701, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. URL https://arxiv.org/abs/2504.10479.

Yuke Zhu, Chi Xie, Shuang Liang, Bo Zheng, and Sheng Guo. FocusLLaVA: A coarse-to-fine approach for efficient and effective visual token compression. *arXiv preprint arXiv:2411.14228*, 2024. URL https://arxiv.org/abs/2411.14228.

Jiedong Zhuang, Lu Lu, Ming Dai, Rui Hu, Jian Chen, Qiang Liu, and Haoji Hu. ST3: Accelerating multimodal large language model by spatial-temporal visual token trimming. In *AAAI*, pp. 11049–11057, 2025. doi: 10.1609/aaai.v39i10.33201. URL https://ojs.aaai.org/index.php/AAAI/article/view/33201.

Appendix

CONTENTS

A Full Related Work **18**

B Algorithms **20**

B.1 Position-Wise Temporal Merging 20

B.2 Request-Aware Spatial Pruning 20

B.3 Sparsity-Aware Attention Compensation 21

C Benchmarks **22**

C.1 MVBench 22

C.2 Video-MME 22

C.3 EgoSchema 22

C.4 LongVideoBench 22

C.5 MLVU 23

D Implementation Details **23**

D.1 Deployment Details 23

D.2 Evaluation Prompts 23

E Additional Experimental Results **26**

E.1 Complete Accuracy Results 26

E.2 Fine-Grained Accuracy by Video Duration 28

E.3 Efficiency Across Retention Ratios 30

E.4 Detailed Component Ablation Results 33

E.5 Detailed Hyperparameter Results 33

F Additional Case Studies **38**

A Full Related Work

Token compression along temporal and spatial dimensions. Visual token compression exploits redundancy within frames and across time (Shao et al., 2026; Yao et al., 2026). Early vision transformers explored learned pooling (Ryoo et al., 2021), adaptive dropping and aggregation (Rao et al., 2021; Liang et al., 2022), and similarity-based merging (Bolya et al., 2023). Multimodal architectures reduce visual sequence length through learned resampling (Alayrac et al., 2022; Li et al., 2023), compact projectors (Cha et al., 2024; Li et al., 2025b), pooling (Yao et al., 2024), or compression tokens (Ye et al., 2025b), sometimes jointly redesigning visual encoding (Vasu et al., 2025; Liu et al., 2025). Training-free methods instead reduce tokens in existing models (Shang et al., 2025; Wang et al., 2025; Arif et al., 2025). VisionZip (Yang et al., 2025b) retains high-attention tokens and merges other features into contextual tokens, preserving salient content and a compact summary without request guidance. Selection criteria also emphasize diversity (Alvar et al., 2025; Wen et al., 2025; Zhang et al., 2025a), input-adaptive policies (Baek et al., 2026; Liang et al., 2026), or importance measures that do not require attention maps (Yang et al., 2025a; Choi et al., 2025; Ma et al., 2026). Videos introduce repeated content across frames, motivating temporal redundancy reduction (Tao et al., 2025; Shao et al., 2025), spatiotemporal clustering (Huang et al., 2025; Hyun et al., 2025), and selection based on segment density or temporal novelty (Shen et al., 2025a; Park et al., 2026). Longer inputs further motivate compression inside vision encoders (Hu et al., 2026; Wang et al., 2026a), hierarchical video processing (Li et al., 2024b), and streaming strategies that suppress redundant incoming tokens (Yao et al., 2025; Chen et al., 2026) or reuse visual features (Wang et al., 2026b). FrameFusion (Fu et al., 2025b) merges spatially aligned tokens from adjacent frames before the first LLM layer and across shallow layers, then prunes by importance. FlashVID (Fan et al., 2026b) combines attention and diversity-based selection with tree-based merging across spatial positions and frames, followed by late-stage LLM pruning. Our temporal merging shares FrameFusion's local temporal matching, but completes merging in one stage before the LLM using compression units with a maximum span. Unlike FlashVID's cross-position merging, it restricts matching to adjacent tokens at the same spatial position. This keeps matching local and limits the temporal extent of averaging. Together with request-aware spatial pruning, it completes token reduction before LLM prefill, so every LLM layer processes the final reduced sequence.

Text-guided token compression. Visual salience alone does not reveal which evidence a question requires. Early approaches incorporated text into token pruning (Cao et al., 2023; 2024) and visual feature extraction (Dai et al., 2023; Li et al., 2024c). Within the LLM, FastV (Chen et al., 2024) ranks visual tokens using early-layer attention and removes low-ranked tokens from subsequent layers. SparseVLM (Zhang et al., 2025d) selects visually relevant text tokens to rate visual tokens, progressively prunes them, and recycles discarded information. Other methods adapt pruning across layers and inputs (Xing et al., 2025; Ye et al., 2025a; Sun et al., 2025; He et al., 2025; Zhuang et al., 2025) or combine vision-side reduction with language-guided filtering (Liu et al., 2024; Zhu et al., 2024). To reduce the sequence before LLM computation, other methods fuse visual information into text (Zhang et al., 2025c) or directly use text–image similarity for selection (Song et al., 2025). Request relevance can also guide diversity-based subset selection (Zhang et al., 2025b) and token merging (Yu et al., 2026). In long and streaming videos, text guidance extends to frame selection, feature reduction, and temporal grounding (Shen et al., 2025b; Zheng et al., 2026; Fan et al., 2026a; Jiang et al., 2026). Closely related to our spatial pruning, AdaFV (Han et al., 2025) combines visual salience and text–image similarity before the LLM by adaptively allocating the token budget between two ranked sets. Our spatial pruning instead assigns each candidate a mixed score, with the mixing weight determined by its original vision attention rank. Its logarithmic schedule gives visual salience greater weight at high ranks and increases the influence of request relevance near the selection boundary. Applied within each frame after temporal merging, this ranking completes the remaining reduction while balancing salient content with request-specific evidence.

Attention distortion after token compression. Selecting informative tokens does not guarantee that their influence on subsequent attention is preserved. Position-aware compression addresses changes to spatial structure (Huang et al., 2026a;b), while attention calibration addresses the changed influence of retained tokens. ToMe (Bolya et al., 2023) adds a logarithmic bias based on merged-token size. RESTORE (Cho et al., 2026) further calibrates attention in multimodal LLMs using both merged-group size and positional distance, while retaining original position indices. Our attention compensation also uses multiplicity-based logit biases, but separates the effects of temporal merging and spatial pruning. It counts the tokens represented by each temporal unit and derives a frame-

wise spatial factor from the ratio of vision attention mass before and after pruning, rather than from positional distance. Using this mass as a proxy for discarded importance makes the spatial correction sensitive to visual salience, instead of treating all removed tokens equally. Both factors are computed before LLM prefill and applied as per-token biases compatible with FlashAttention (Dao et al., 2022; Dao, 2024; Shah et al., 2024; Zadouri et al., 2026), linking the correction to efficient inference.

B ALGORITHMS

Algorithms 1–3 describe temporal merging, spatial pruning, and attention compensation. They follow the notation in Section 2. Before temporal merging, we compute normalized request relevance on the original visual tokens using Equation 7; this allows both attention and relevance scores to be averaged over exactly the same compression units as the embeddings.

B.1 POSITION-WISE TEMPORAL MERGING

Algorithm 1 applies the similarity, maximum-span, and merge-budget constraints before averaging each final unit. A singleton has temporal multiplicity one. Each merged representative retains its first source token's position, and its attention mass is the sum of the source attention scores, equivalently the mean score multiplied by the unit size. On Qwen3-VL, all DeepStack feature levels follow the same units and selection indices. The token grid follows the vision encoder's native temporal grouping. Model-specific separator tokens are reserved separately when converting the target retention ratio into the compressible visual token budget.

Algorithm 1 Position-wise temporal merging

Input: Visual embeddings $\mathbf{E}_v$, attention scores a, normalized relevance scores q;
similarity threshold τ, maximum span $s_{\max}$, target token count K.
Output: Merged candidates with root positions, averaged scores, and temporal multiplicities.
Identify and split temporal units (Equations 2–5).
Compute $S_{n,p} \leftarrow \cos(\mathbf{e}_{n,p}, \mathbf{e}_{n+1,p})$ for all adjacent-frame pairs.
for each spatial position p **do**
  $\mathcal{B}^p \leftarrow \{0, N\} \cup \{n \mid S_{n,p} \le \tau\}$.
  Form the unit list from consecutive boundaries in $\mathcal{B}^p$.
  for each unit U in this list **do**
    $c \leftarrow \lceil |U| / s_{\max} \rceil$.
    Split U into c contiguous parts differing in size by at most one, longer parts first.
    Add the internal split boundaries to $\mathcal{B}^p$.
  end for
end for
Enforce the global merge budget (Equation 6).
$M \leftarrow \sum_p (|\mathcal{B}^p| - 1)$.
if $M < K$ **then**
  Select the $K - M$ lowest-similarity connections not already cut by a boundary.
  Add each selected connection (n, p) as boundary n in $\mathcal{B}^p$.
end if
Merge features and scores; preserve source-token counts (Equation 4).
Initialize $\mathcal{C}_n \leftarrow \emptyset$ for every frame n.
for each final unit U at position p **do**
  Let i be the original flattened index of its first token and n its frame.
  $(\mathbf{e}_i, a_i, q_i) \leftarrow$ component-wise averages of the embeddings and scores in U.
  $m_i^{\mathrm{t}} \leftarrow |U|$; add i to $\mathcal{C}_n$ and discard the other tokens in U.
end for
return $\{\mathcal{C}_n\}_n$ and the retained features, scores, positions, and m^{t}.

B.2 REQUEST-AWARE SPATIAL PRUNING

Algorithm 2 first assigns an exact frame-wise budget. If K is at least the number of nonempty frames, ALLOCATEBUDGETS reserves one token per nonempty frame and distributes the remaining budget in proportion to the residual candidate counts. Otherwise, it allocates the entire budget in proportion to the candidate counts. In either case, it rounds allocations down and assigns the remaining slots by descending fractional remainder, breaking ties by frame order. This preserves the global budget without exceeding any frame's capacity.

The schedule g makes the configuration explicit. For a candidate at attention rank $j < K_n$, C1 uses $g(j, K_n) = \log(K_n - j + 1)/\log(K_n + 1)$, as in Equation 8. C2 uses $g(j, K_n) = 1 - \log(j + 1)/\log(K_n + 1)$. The linear ablation uses $g(j, K_n) = (K_n - j)/K_n$. All schedules use α_{tail} beyond the selection boundary. The initial attention sort is stable, and selected tokens are gathered in their original order.

Algorithm 2 Request-aware spatial pruning

Input: Temporal candidates $\{\mathcal{C}_n\}_n$, averaged scores a_i, q_i, target count K;
weighting endpoints $\alpha_{\text{head}}, \alpha_{\text{tail}}$ and schedule g.
Output: Selected sets $\{\mathcal{C}_n^\star\}_n$ with exactly K visual tokens.

Allocate the remaining budget within frames.
$K_n \leftarrow \text{AllocateBudgets}(\{|\mathcal{C}_n|\}_n, K)$ for every frame n.
for each frame n **do**
  $\mathcal{C}_n^\star \leftarrow \emptyset$.
  if $K_n = 0$ **then continue**.
  if $K_n = |\mathcal{C}_n|$ **then**
    $\mathcal{C}_n^\star \leftarrow \mathcal{C}_n$; **continue**.
  end if
  Weight request relevance by the original attention rank (Equations 8–9).
  $(\pi_0, \ldots, \pi_{|\mathcal{C}_n|-1}) \leftarrow$ candidate indices sorted by decreasing a_i.
  for $j = 0, \ldots, |\mathcal{C}_n| - 1$ **do**
    $\alpha \leftarrow \alpha_{\text{tail}}$.
    if $j < K_n$ **then**
      $\alpha \leftarrow \alpha_{\text{tail}} + (\alpha_{\text{head}} - \alpha_{\text{tail}})g(j, K_n)$.
    end if
    $\text{score}_{\pi_j} \leftarrow \alpha a_{\pi_j} + (1 - \alpha)q_{\pi_j}$.
  end for
  $\mathcal{C}_n^\star \leftarrow \text{TopK}((\text{score}_i)_{i \in \mathcal{C}_n}, K_n)$.
end for
Gather the selected tokens and their metadata in the original token order.
return $\{\mathcal{C}_n^\star\}_n$ and the selected features, positions, and multiplicities.

B.3 Sparsity-Aware Attention Compensation

Algorithm 3 constructs one bias for each retained visual token. Nonvisual tokens receive zero bias. If a frame has invalid or zero retained attention mass, the implementation omits its spatial correction rather than taking an undefined logarithm. Compensation is applied during LLM prefill only and is disabled during decoding.

The implementation uses the original attention scale s. An auxiliary query coordinate of 1 and key coordinate of b_i/s add exactly b_i to the scaled dot product. Query, key, and value tensors are padded to the same supported head dimension; auxiliary value coordinates remain zero, and the corresponding output coordinates are discarded. The original causal and padding masks are preserved. This realizes the additive bias without materializing the full attention logit matrix or replacing FlashAttention.

Algorithm 3 Sparsity-aware attention compensation during LLM prefill

Input: Candidate and selected sets $\mathcal{C}_n, \mathcal{C}_n^\star$, scores a_i, counts m_i^{t};
query, key, and value tensors $\mathbf{Q}, \mathbf{K}, \mathbf{V}$, original scale s and attention mask.
Output: Attention output with temporal and spatial compensation.

Construct a bias for each retained visual token (Equations 11–12).

Initialize $b_i \leftarrow 0$ for every token in the LLM input.
for each frame n with $\mathcal{C}_n^\star \neq \emptyset$ **do**
  $A_{\mathrm{all}} \leftarrow \sum_{j \in \mathcal{C}_n} m_j^{\mathrm{t}} a_j$, $A_{\mathrm{keep}} \leftarrow \sum_{j \in \mathcal{C}_n^\star} m_j^{\mathrm{t}} a_j$.
  $m_n^{\mathrm{s}} \leftarrow 1$.
  if all candidate masses are finite and nonnegative and $A_{\mathrm{keep}} > 0$ **then**
    $m_n^{\mathrm{s}} \leftarrow \max(1, A_{\mathrm{all}}/A_{\mathrm{keep}})$.
  end if
  for $i \in \mathcal{C}_n^\star$ **do**
    $b_i \leftarrow \log m_i^{\mathrm{t}} + \log m_n^{\mathrm{s}}$.
  end for
end for

Introduce the bias through an auxiliary query–key coordinate.

Let d be the original head dimension; choose a supported augmented dimension $d' > d$.
Zero-pad $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ to d' dimensions, obtaining $\widetilde{\mathbf{Q}}, \widetilde{\mathbf{K}}, \widetilde{\mathbf{V}}$.
Set the first auxiliary query coordinate to 1 at every valid query position.
Set the first auxiliary key coordinate to b_i/s at key position i, for every head.
$\widetilde{\mathbf{O}} \leftarrow \mathrm{FlashAttention}(\widetilde{\mathbf{Q}}, \widetilde{\mathbf{K}}, \widetilde{\mathbf{V}};\ \mathrm{scale} = s, \mathrm{original\ mask})$.
return the first d coordinates of $\widetilde{\mathbf{O}}$.

C BENCHMARKS

C.1 MVBENCH

MVBench (Li et al., 2024a) evaluates temporal understanding across 20 tasks, including action sequencing, motion perception, and object interaction. These tasks require information from multiple frames rather than a single image. We evaluate all 4,000 multiple-choice questions, with 200 questions per task, and report overall accuracy.

C.2 VIDEO-MME

Video-MME (Fu et al., 2025a) evaluates video understanding across six domains and 30 subcategories. It contains 2,700 multiple-choice questions on 900 videos, ranging from 11 seconds to one hour and divided into short, medium, and long groups. We evaluate without subtitles or audio and report overall accuracy across all questions.

C.3 EGOSCHEMA

EgoSchema (Mangalam et al., 2023) tests understanding of everyday activities in three-minute egocentric videos. Each question has five answer choices and requires reasoning over events across the video. We evaluate the full set of 5,031 questions, rather than the 500-question subset, and report accuracy from the official evaluation server.

C.4 LONGVIDEOBENCH

LongVideoBench (Wu et al., 2024) evaluates retrieval and reasoning over videos up to one hour long. Its questions cover 17 categories and require models to locate the context described in a question, identify relevant details, or relate events across scenes. We use the validation set of 1,337 multiple-choice questions, without subtitles or audio, and report overall accuracy.

C.5 MLVU

MLVU (Zhou et al., 2025) assesses long video understanding at the level of overall content, individual details, and relations between events. We evaluate its seven multiple-choice tasks on the development set (2,174 questions): topic reasoning, anomaly recognition, needle QA, ego reasoning, plot QA, action order, and action count. We report M-Avg, the unweighted mean of accuracy across these tasks, and exclude the free-form generation tasks.

D Implementation Details

D.1 Deployment Details

Baseline deployment. We retain each baseline's compression design and report its budget convention. Saevia and FastV report the final visual token fraction; SparseVLM and FlashVID report an equivalent layer-averaged visual token budget. VisionZip compresses before the LLM, so its retained sequence is shared across language layers. FastV prunes after block 2 on LLaVA and block 3 on Qwen3-VL, after DeepStack feature injection. SparseVLM uses zero-based layer indices 2, 6, and 15. FlashVID combines vision-side compression with LLM pruning after block 20 on LLaVA and block 28 on Qwen3-VL. All methods share model-specific frame samples, prompts, and decoding settings. We reset the random seed to 3407 for each request. Qwen3-VL and LLaVA-OneVision use BF16, while LLaVA-Video uses FP16; no model is quantized.

Saevia deployment. Saevia compresses visual features before LLM prefill. On Qwen3-VL, the final visual embeddings and all DeepStack feature levels undergo the same temporal merging and spatial selection. The compressed DeepStack features therefore remain aligned with the retained visual tokens when injected into the LLM. Retained tokens keep their original position IDs, with each merged unit represented at its first token's position. On LLaVA-OneVision, we apply compression after the model's bilinear spatial downsampling and downsample the attention scores to the same token grid. Omitting a forced contiguous conversion preserves the channels-last memory layout for efficient CUDA downsampling. The final vision layer returns its existing queries and keys, from which we reconstruct attention scores while retaining FlashAttention in the vision encoder. For Qwen3-VL performance profiling, a Triton backend computes the required softmax statistics and scores in a bounded workspace. Spatial pruning batches ranking and selection in PyTorch, with Triton kernels for score packing, index compaction, and feature gathering. Temporal merging uses PyTorch. Attention compensation appends auxiliary query, key, and value coordinates to introduce the logit bias through the dot product, without replacing FlashAttention. It is enabled only during prefill. Appendix B specifies the compression algorithms and the auxiliary-coordinate construction.

D.2 Evaluation Prompts

The prompts are grouped by model: LLaVA-OneVision, LLaVA-Video, and Qwen3-VL. Benchmarks with the same instruction template share a panel. The two LLaVA models use the same benchmark-specific templates, shown separately for ease of reference. Qwen3-VL shares one template across MVBench, Video-MME, EgoSchema, and MLVU, with a separate template for LongVideoBench. Within each model and benchmark, the full-token baseline and all compression methods use identical prompts. The model processor applies its native chat template and video-input formatting to the text shown below.

The placeholders {`question`}, {`options`}, and {`letters`} denote the question, answer choices listed one per line, and valid option letters, respectively. For example, four choices give `A, B, C, or D`. MVBench and MLVU use parenthesized option labels such as `(A)`, while LongVideoBench uses labels such as `A.`; Video-MME and EgoSchema preserve the option strings supplied by the datasets. Line breaks and spacing are adjusted for readability. No audio or subtitles are supplied. The LLaVA Video-MME template retains its original reference to subtitles.

D.2.1 LLaVA-OneVision

MVBench

```
Question: {question}
Option:
{options}
Only give the best option ({letters}).
```

Video-MME$_{w/o\ sub.}$

```
Select the best answer to the following multiple-choice question
based on the video and the subtitles.
Respond with only the letter ({letters}) of the correct option.
{question}
{options}
Answer with the option's letter from the given choices directly.
```

EgoSchema$_{full}$ and LongVideoBench$_{val.}$

```
{question}
{options}
Answer with the option's letter ({letters}) from the given
choices directly.
```

MLVU$^{Dev}_{M-Avg}$

```
{question}
{options}
Only give the best option ({letters}).
Best option: (
```

D.2.2 LLaVA-Video

MVBench

```
Question: {question}
Option:
{options}
Only give the best option ({letters}).
```

Video-MME$_{w/o\ sub.}$

```
Select the best answer to the following multiple-choice question
based on the video and the subtitles.
Respond with only the letter ({letters}) of the correct option.
{question}
{options}
Answer with the option's letter from the given choices directly.
```

EgoSchema$_{full}$ and LongVideoBench$_{val.}$

```
{question}
{options}
Answer with the option's letter ({letters}) from the given
choices directly.
```

MLVU$^{Dev}_{M\text{-}Avg}$

```
{question}
{options}
Only give the best option ({letters}).
Best option:  (
```

D.2.3 QWEN3-VL

MVBench, Video-MME$_{w/o\ sub.}$, EgoSchema$_{full}$, and MLVU$^{Dev}_{M\text{-}Avg}$

```
<video>
Select the best answer to the following multiple-choice question
based on the video.
Respond with only the letter ({letters}) of the correct option.
Question:  {question}
Possible answer choices:
{options}
The best answer is:
```

LongVideoBench$_{val.}$

```
Select the best answer to the following multiple-choice question
based on the video.
Respond with only the letter ({letters}) of the correct option.
Question:  {question}
{options}
The best answer is:
```

E Additional Experimental Results

Accuracy and efficiency results use C1 for LLaVA-OneVision and LLaVA-Video, and C2 for Qwen3-VL, matching the main experiments. Hyperparameter sensitivity uses C1 for all three models. Qwen3-VL accuracy results use 128 uniformly sampled frames; the fixed performance workload follows Section 3.2 and Appendix E.3. For the accuracy, efficiency, and hyperparameter comparisons, we report the complete five-point retention sweep rather than selecting individual winning settings. Across the tables, Avg. is the unweighted mean of the five benchmark scores, and Pct. expresses that average relative to the corresponding full-token baseline. Within each model and retention ratio, the **best** and second-best results among compressed variants are shown in bold and underlined. Ties at the reported precision receive the same rank.

E.1 Complete Accuracy Results

Tables 5–7 report complete accuracy results at all five retention ratios for LLaVA-OneVision, LLaVA-Video, and Qwen3-VL. Saevia achieves the highest average accuracy at all five ratios on both LLaVA models. On Qwen3-VL, it leads at 10–20% retention but not at 25–30%, showing that the relative benefit depends on the token budget.

Table 5: Complete accuracy results for **LLaVA-OneVision-7B**.

Method	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
FastV	30%	54.17	57.19	56.63	54.23	59.00	56.24	97.65
VisionZip	30%	55.07	58.85	57.98	54.53	61.26	57.54	99.90
SparseVLM	30%	54.87	57.56	57.70	51.83	60.06	56.40	97.93
FlashVID	30%	**55.47**	**59.07**	57.98	55.35	61.35	57.84	100.43
Saevia	30%	55.37	58.89	**59.07**	**55.50**	**61.50**	**58.07**	**100.82**
FastV	25%	54.10	56.37	56.21	52.88	58.54	55.62	96.57
VisionZip	25%	54.85	58.26	57.98	54.75	61.07	57.38	99.63
SparseVLM	25%	54.57	57.41	56.93	50.79	60.17	55.97	97.18
FlashVID	25%	55.05	**59.30**	58.26	**54.97**	61.41	57.80	100.35
Saevia	25%	**55.50**	58.96	**59.75**	54.90	**62.01**	**58.22**	**101.08**
FastV	20%	53.10	56.48	55.89	52.06	57.79	55.06	95.60
VisionZip	20%	54.75	58.07	57.74	53.33	60.26	56.83	98.67
SparseVLM	20%	53.75	55.78	56.45	48.17	59.04	54.64	94.86
FlashVID	20%	54.65	**58.48**	57.94	**54.82**	61.50	57.48	99.80
Saevia	20%	**55.67**	58.15	**59.87**	54.00	**62.09**	**57.96**	**100.63**
FastV	15%	52.90	54.89	54.84	51.38	56.63	54.13	93.98
VisionZip	15%	53.53	55.96	57.13	52.66	58.79	55.61	96.56
SparseVLM	15%	51.75	52.74	53.81	47.05	56.20	52.31	90.82
FlashVID	15%	54.65	58.04	57.88	**56.17**	60.67	57.48	99.80
Saevia	15%	**55.60**	**58.11**	**60.03**	54.53	**61.41**	**57.94**	**100.59**
FastV	10%	51.88	52.30	53.35	46.82	54.16	51.70	89.77
VisionZip	10%	51.85	52.74	55.10	47.57	57.11	52.87	91.80
SparseVLM	10%	41.88	44.26	39.67	42.63	45.76	42.84	74.38
FlashVID	10%	53.73	**56.59**	57.36	53.93	59.78	56.28	97.71
Saevia	10%	**55.05**	56.37	**59.87**	**54.23**	**60.22**	**57.15**	**99.22**

Table 6: Complete accuracy results for **LLaVA-Video-7B**.

Method	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
FastV	30%	55.57	61.41	54.56	54.75	60.91	57.44	95.81
VisionZip	30%	**56.90**	62.59	56.29	56.84	64.61	59.45	99.17
SparseVLM	30%	55.75	61.96	54.72	54.82	61.83	57.82	96.45
FlashVID	30%	55.97	**63.37**	56.11	56.39	64.81	59.33	98.97
Saevia	30%	55.75	61.78	**56.99**	**57.59**	**65.42**	**59.51**	**99.27**
FastV	25%	55.27	60.59	54.50	53.85	60.50	56.94	94.98
VisionZip	25%	**56.93**	61.70	56.51	56.39	63.93	59.09	98.57
SparseVLM	25%	55.20	60.74	54.08	54.00	60.57	56.92	94.95
FlashVID	25%	55.75	**62.89**	55.50	56.77	64.18	59.02	98.45
Saevia	25%	55.83	62.48	**57.46**	**57.59**	**65.20**	**59.71**	**99.60**
FastV	20%	54.63	59.19	53.57	53.10	59.15	55.93	93.29
VisionZip	20%	**56.45**	62.00	56.03	55.87	63.12	58.69	97.90
SparseVLM	20%	54.15	59.44	52.65	52.13	58.99	55.47	92.53
FlashVID	20%	55.90	**62.11**	55.77	**56.77**	63.38	58.79	98.07
Saevia	20%	55.93	61.89	**57.13**	56.17	**65.31**	**59.29**	**98.90**
FastV	15%	54.23	58.93	52.40	51.23	58.48	55.05	91.83
VisionZip	15%	**56.23**	60.78	55.04	53.78	63.13	57.79	96.40
SparseVLM	15%	52.40	56.52	49.41	47.57	56.32	52.44	87.47
FlashVID	15%	55.87	61.52	55.83	**55.95**	63.94	58.62	97.78
Saevia	15%	55.50	**61.89**	**56.75**	55.42	**64.53**	**58.82**	**98.12**
FastV	10%	53.20	55.89	50.31	50.79	57.80	53.60	89.41
VisionZip	10%	54.75	58.78	52.83	52.66	60.56	55.92	93.28
SparseVLM	10%	43.87	46.63	35.84	43.83	48.70	43.77	73.01
FlashVID	10%	**55.77**	60.19	54.60	54.23	62.39	57.44	95.81
Saevia	10%	54.65	**60.93**	**55.85**	**54.90**	**62.85**	**57.84**	**96.48**

Table 7: Complete accuracy results for **Qwen3-VL-8B-Instruct**.

Method	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
FastV	30%	66.20	67.48	71.81	62.90	68.82	67.44	97.70
VisionZip	30%	**66.27**	**68.52**	**72.35**	63.35	**70.87**	**68.27**	**98.90**
SparseVLM	30%	65.80	68.22	71.89	**63.87**	69.94	67.94	98.43
FlashVID	30%	64.68	67.78	71.40	63.58	69.22	67.33	97.54
Saevia	30%	64.92	67.56	71.20	63.65	69.66	67.40	97.64
FastV	25%	**66.17**	67.22	71.54	62.90	67.75	67.12	97.23
VisionZip	25%	66.03	**68.48**	**71.89**	64.10	**69.63**	**68.03**	**98.55**
SparseVLM	25%	64.95	67.59	71.48	**64.17**	69.26	67.49	97.77
FlashVID	25%	64.12	67.33	71.40	63.28	68.42	66.91	96.93
Saevia	25%	64.98	66.89	71.40	64.02	69.60	67.38	97.61
FastV	20%	**66.20**	66.11	71.18	62.00	65.39	66.18	95.87
VisionZip	20%	65.35	**67.59**	**71.91**	63.05	68.27	67.23	97.39
SparseVLM	20%	64.98	66.96	71.02	61.41	67.37	66.35	96.11
FlashVID	20%	64.52	66.59	71.18	63.50	68.20	66.80	96.76
Saevia	20%	64.85	67.22	71.40	**63.95**	**68.93**	**67.27**	**97.45**
FastV	15%	**65.18**	65.48	70.28	60.43	63.09	64.89	94.00
VisionZip	15%	64.62	67.00	71.06	62.98	66.52	66.44	96.25
SparseVLM	15%	63.65	65.30	70.72	61.93	64.22	65.16	94.40
FlashVID	15%	63.85	66.04	71.02	62.98	66.85	66.15	95.82
Saevia	15%	64.80	**67.04**	**71.18**	**63.05**	**68.44**	**66.90**	**96.91**
FastV	10%	63.88	63.78	69.51	58.56	60.01	63.15	91.48
VisionZip	10%	61.30	64.00	69.81	59.99	65.30	64.08	92.83
SparseVLM	10%	56.17	59.93	65.59	52.28	53.63	57.52	83.32
FlashVID	10%	62.90	65.48	70.18	**62.98**	**66.75**	65.66	95.11
Saevia	10%	**64.30**	**66.15**	**70.68**	61.33	66.48	**65.79**	**95.31**

E.2 Fine-Grained Accuracy by Video Duration

Tables 8–10 report accuracy across video-duration groups on Video-MME and LongVideoBench for all three models. They include all five retention ratios under the same configurations as the main results.

Table 8: Accuracy results by video duration for **LLaVA-OneVision-7B**.

Method	Ret. Ratio	Video-MME$_{w/o\ sub.}$				LongVideoBench$_{val.}$				
		Overall	Short	Medium	Long	Overall	8–15s	15–60s	180–600s	900–3600s
Full-token baseline	100%	58.33	70.22	56.22	48.56	54.23	65.08	69.19	51.94	47.70
FastV	30%	57.19	69.44	54.78	47.33	54.23	65.61	68.60	51.94	47.70
VisionZip	30%	58.85	70.33	**57.11**	49.11	54.53	67.20	69.77	**53.88**	46.10
SparseVLM	30%	57.56	69.11	56.22	47.33	51.83	65.08	64.53	49.03	45.57
FlashVID	30%	**59.07**	**71.56**	56.44	49.22	55.35	67.20	69.77	52.67	**48.94**
Saevia	30%	58.89	70.33	56.00	**50.33**	**55.50**	**67.72**	**70.93**	53.64	48.05
FastV	25%	56.37	67.22	55.22	46.67	52.88	66.67	64.53	50.97	46.10
VisionZip	25%	58.26	69.22	56.67	48.89	54.75	66.14	**69.77**	52.67	47.87
SparseVLM	25%	57.41	67.89	55.89	48.44	50.79	62.96	63.95	48.79	44.15
FlashVID	25%	**59.30**	**70.78**	**57.67**	**49.44**	**54.97**	65.08	**69.77**	51.94	**49.29**
Saevia	25%	58.96	70.11	57.33	**49.44**	54.90	**67.20**	**69.77**	**55.34**	45.92
FastV	20%	56.48	66.33	55.67	47.44	52.06	60.85	65.70	50.73	45.92
VisionZip	20%	58.07	69.11	56.22	48.89	53.33	**66.67**	68.02	53.40	44.33
SparseVLM	20%	55.78	65.11	54.78	47.44	48.17	60.85	62.21	45.63	41.49
FlashVID	20%	**58.48**	**69.78**	**57.22**	48.44	**54.82**	65.61	**69.19**	53.16	**48.05**
Saevia	20%	58.15	69.22	56.22	**49.00**	54.00	66.14	68.02	**53.64**	45.92
FastV	15%	54.89	64.89	52.67	47.11	51.38	61.38	66.28	49.03	45.21
VisionZip	15%	55.96	64.89	54.89	48.11	52.66	63.49	65.70	53.16	44.68
SparseVLM	15%	52.74	60.78	51.67	45.78	47.05	57.67	59.30	43.93	42.02
FlashVID	15%	58.04	68.67	**57.11**	48.33	**56.17**	**67.72**	**72.67**	**55.34**	**47.87**
Saevia	15%	**58.11**	**69.11**	56.33	**48.89**	54.53	67.20	67.44	54.85	46.10
FastV	10%	52.30	59.78	52.33	44.78	46.82	57.14	58.72	43.93	41.84
VisionZip	10%	52.74	59.11	51.78	47.33	47.57	55.56	55.23	49.03	41.49
SparseVLM	10%	44.26	46.11	44.44	42.22	42.63	43.92	48.26	42.72	40.43
FlashVID	10%	**56.59**	**67.33**	54.11	48.33	53.93	62.43	**69.19**	**55.58**	45.21
Saevia	10%	56.37	66.11	**54.56**	**48.44**	**54.23**	**68.25**	66.28	53.64	**46.28**

Table 9: Accuracy results by video duration for **LLaVA-Video-7B**.

Method	Ret. Ratio	Video-MME$_{w/o\ sub.}$				LongVideoBench$_{val.}$				
		Overall	Short	Medium	Long	Overall	8–15s	15–60s	180–600s	900–3600s
Full-token baseline	100%	63.85	76.33	61.00	54.22	56.54	67.20	69.77	55.34	49.82
FastV	30%	61.41	72.78	59.44	52.00	54.75	65.08	69.77	55.34	46.28
VisionZip	30%	62.59	75.22	60.22	52.33	56.84	69.31	71.51	**56.55**	48.40
SparseVLM	30%	61.96	73.89	59.11	**52.89**	54.82	63.49	70.93	55.10	46.81
FlashVID	30%	**63.37**	**75.44**	**61.89**	52.78	56.39	67.20	**74.42**	53.88	49.11
Saevia	30%	61.78	72.44	60.44	52.44	**57.59**	**70.37**	70.93	54.61	**51.42**
FastV	25%	60.59	71.89	58.78	51.11	53.85	65.08	66.86	53.88	46.10
VisionZip	25%	61.70	72.89	60.00	52.22	56.39	68.25	73.84	**56.55**	46.99
SparseVLM	25%	60.74	72.67	58.56	51.00	54.00	62.43	67.44	54.37	46.81
FlashVID	25%	**62.89**	**75.00**	**60.89**	52.78	56.77	68.25	**74.42**	54.37	49.29
Saevia	25%	62.48	74.11	60.22	**53.11**	**57.59**	**68.78**	72.67	55.58	**50.71**
FastV	20%	59.19	69.44	58.11	50.00	53.10	65.08	65.70	52.91	45.39
VisionZip	20%	62.00	72.11	**60.44**	**53.44**	55.87	65.61	69.77	**56.55**	47.87
SparseVLM	20%	59.44	70.22	57.78	50.33	52.13	62.96	64.53	52.43	44.50
FlashVID	20%	**62.11**	**74.11**	60.22	52.00	**56.77**	67.72	**75.00**	54.85	**48.94**
Saevia	20%	61.89	72.89	60.11	52.67	56.17	**69.84**	73.84	52.67	48.76
FastV	15%	58.93	68.22	57.67	50.89	51.23	61.90	65.12	50.97	43.62
VisionZip	15%	60.78	71.78	58.78	51.78	53.78	60.32	68.60	53.64	47.16
SparseVLM	15%	56.52	65.33	55.22	49.00	47.57	56.08	59.30	47.09	41.49
FlashVID	15%	61.52	72.56	59.44	52.56	**55.95**	**66.67**	**74.42**	**54.85**	47.52
Saevia	15%	**61.89**	**73.22**	**59.67**	**52.78**	55.42	66.14	73.26	53.64	**47.70**
FastV	10%	55.89	64.33	54.00	49.33	50.79	61.90	62.79	49.76	44.15
VisionZip	10%	58.78	68.22	57.00	51.11	52.66	60.32	61.63	54.13	46.28
SparseVLM	10%	46.63	50.44	46.67	42.78	43.83	47.09	51.16	43.93	40.43
FlashVID	10%	60.19	71.00	57.78	**51.78**	54.23	65.08	**69.77**	**54.37**	45.74
Saevia	10%	**60.93**	**72.33**	**59.89**	50.56	**54.90**	**67.72**	67.44	**54.37**	**47.16**

Table 10: Accuracy results by video duration for **Qwen3-VL-8B-Instruct**.

Method	Ret. Ratio	Video-MME$_{w/o\ sub.}$				LongVideoBench$_{val.}$				
		Overall	Short	Medium	Long	Overall	8–15s	15–60s	180–600s	900–3600s
Full-token baseline	100%	69.11	80.33	68.44	58.56	63.95	79.89	75.58	63.11	55.67
FastV	30%	67.48	**78.56**	66.56	57.33	62.90	79.89	75.58	62.62	53.55
VisionZip	30%	**68.52**	78.11	67.89	**59.56**	63.35	**80.95**	73.84	64.56	53.37
SparseVLM	30%	68.22	78.44	**68.22**	58.00	**63.87**	80.42	**77.91**	63.83	54.08
FlashVID	30%	67.78	78.00	67.67	57.67	63.58	78.84	75.58	63.35	**54.96**
Saevia	30%	67.56	78.00	66.00	58.67	63.65	78.31	77.33	**64.81**	53.72
FastV	25%	67.22	**77.78**	66.22	57.67	62.90	79.89	75.00	63.35	53.19
VisionZip	25%	**68.48**	77.56	**68.11**	**59.78**	64.10	80.42	75.00	**64.81**	**54.79**
SparseVLM	25%	67.59	**77.78**	66.78	58.22	**64.17**	**80.95**	**77.91**	64.56	54.08
FlashVID	25%	67.33	77.67	66.44	57.89	63.28	77.78	75.58	63.59	54.43
Saevia	25%	66.89	76.44	66.44	57.78	64.02	78.84	76.16	**64.81**	**54.79**
FastV	20%	66.11	75.78	64.78	57.78	62.00	79.37	70.35	62.14	53.55
VisionZip	20%	**67.59**	76.78	**66.56**	**59.44**	63.05	**79.89**	75.58	63.59	53.19
SparseVLM	20%	66.96	76.56	65.89	58.44	61.41	76.19	76.16	61.41	51.95
FlashVID	20%	66.59	**77.00**	65.22	57.56	63.50	78.31	75.58	63.35	**54.96**
Saevia	20%	67.22	76.89	66.11	58.67	**63.95**	78.31	**76.74**	**64.81**	54.61
FastV	15%	65.48	75.22	63.33	57.89	60.43	77.25	72.09	61.89	50.18
VisionZip	15%	67.00	76.11	**66.22**	**58.67**	62.98	79.37	76.16	63.59	53.01
SparseVLM	15%	65.30	73.89	64.56	57.44	61.93	79.37	73.84	61.89	52.48
FlashVID	15%	66.04	**77.00**	64.33	56.78	62.98	**79.89**	75.58	63.35	**53.19**
Saevia	15%	**67.04**	76.56	**66.22**	58.33	**63.05**	**79.89**	**76.74**	**64.08**	52.48
FastV	10%	63.78	70.89	62.78	**57.67**	58.56	74.07	69.19	59.47	49.47
VisionZip	10%	64.00	73.00	62.56	56.44	59.99	75.13	75.58	59.22	50.71
SparseVLM	10%	59.93	66.11	58.44	55.22	52.28	66.67	63.37	52.18	44.15
FlashVID	10%	65.48	75.78	63.33	57.33	**62.98**	**80.42**	**76.74**	**63.35**	**52.66**
Saevia	10%	**66.15**	**76.67**	**65.33**	56.44	61.33	77.78	75.58	61.41	51.42

E.3 Efficiency Across Retention Ratios

Tables 11–15 report efficiency at 10%, 15%, 20%, 25%, and 30% retention with the same metrics and presentation as Table 3. The 10% results are repeated here for ease of reference. All measurements use the same real-world 720p video, one NVIDIA A100-SXM4-80GB GPU, one warm-up run, and three measured runs. The workloads sample 32 frames for LLaVA-OneVision and 64 for LLaVA-Video and Qwen3-VL, with approximately 6K, 11K, and 32K input visual tokens, respectively. Each run generates 1,000 tokens. These fixed workloads isolate inference performance under controlled inputs; they are not benchmark-wide latency estimates.

Prefill throughput is the original input token count divided by TTFT, so it measures effective input processing rather than the physical rate of retained tokens through the LLM. End-to-end throughput divides all 1,000 generated tokens by total generation time. Overall TTFT includes residual overhead and therefore need not equal the sum of the three reported stages. Accuracy is taken from the separate five-benchmark evaluation, not from the profiling video.

Table 11: Efficiency results at **10% visual token retention**.

Method		Ret. Ratio	TTFT (ms) ↓				Throughput (tokens/s) ↑		Accuracy	
			Overall	Vision Enc.	Token Proc.	LLM Prefill	Prefill	End-to-End	Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	10%	277.23 (2.29×)	117.41	62.71	96.97	22,960 (2.29×)	40.20 (1.03×)	51.70	89.77
VisionZip	CVPR'25	10%	293.91 (2.16×)	124.95	108.61	**60.23**	21,657 (2.16×)	40.02 (1.03×)	52.87	91.80
SparseVLM	ICML'25	10%	274.90 (2.31×)	116.77	63.00	95.00	23,154 (2.31×)	40.00 (1.02×)	42.84	74.38
FlashVID	ICLR'26 Oral	10%	318.65 (1.99×)	124.58	120.81	73.14	19,975 (1.99×)	**40.34** (1.03×)	56.28	97.71
Saevia	Ours	10%	**199.95** (3.17×)	**116.35**	**21.55**	61.94	**31,833** (3.17×)	40.11 (1.03×)	**57.15**	**99.22**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	10%	498.62 (2.43×)	340.37	4.66	153.42	22,029 (2.43×)	40.06 (1.02×)	53.60	89.41
VisionZip	CVPR'25	10%	497.09 (2.43×)	353.30	53.75	**89.87**	22,097 (2.43×)	40.12 (1.02×)	55.92	93.28
SparseVLM	ICML'25	10%	488.00 (2.48×)	340.13	**4.61**	143.08	22,508 (2.48×)	**41.27** (1.05×)	43.77	73.01
FlashVID	ICLR'26 Oral	10%	539.33 (2.24×)	350.34	85.53	103.28	20,367 (2.24×)	40.72 (1.03×)	57.44	95.81
Saevia	Ours	10%	**461.97** (2.62×)	**338.88**	32.24	90.69	**23,777** (2.62×)	40.80 (1.03×)	**57.84**	**96.48**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	10%	2052.80 (2.65×)	**1429.02**	27.54	596.05	15,791 (2.65×)	25.51 (1.23×)	63.15	91.48
VisionZip	CVPR'25	10%	2118.89 (2.57×)	1599.48	231.41	**287.82**	15,298 (2.57×)	25.80 (1.25×)	64.08	92.83
SparseVLM	ICML'25	10%	1951.85 (2.79×)	1438.38	**23.82**	489.45	16,607 (2.79×)	25.29 (1.22×)	57.52	83.32
FlashVID	ICLR'26 Oral	10%	2042.98 (2.67×)	1610.18	120.78	311.84	15,867 (2.67×)	25.23 (1.22×)	65.66	95.11
Saevia	Ours	10%	**1873.54** (2.91×)	1450.93	122.14	300.32	**17,302** (2.91×)	**25.91** (1.25×)	**65.79**	**95.31**

Table 12: Efficiency results at **15% visual token retention**.

Method		Ret. Ratio	TTFT (ms) ↓ Overall	Vision Enc.	Token Proc.	LLM Prefill	Throughput (tokens/s) ↑ Prefill	End-to-End	Accuracy Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	15%	300.79 (2.11×)	118.14	63.30	119.21	21,161 (2.11×)	39.61 (1.01×)	54.13	93.98
VisionZip	CVPR'25	15%	310.48 (2.04×)	125.50	99.23	**85.62**	20,500 (2.04×)	40.58 (1.04×)	55.61	96.56
SparseVLM	ICML'25	15%	285.69 (2.22×)	117.56	63.00	104.99	22,280 (2.22×)	39.92 (1.02×)	52.31	90.82
FlashVID	ICLR'26 Oral	15%	338.09 (1.88×)	124.52	125.98	87.48	18,830 (1.88×)	40.72 (1.04×)	57.48	99.80
Saevia	Ours	15%	**226.20** (2.80×)	**117.53**	**20.88**	87.68	**28,139** (2.80×)	**40.73** (1.04×)	**57.94**	**100.59**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	15%	539.60 (2.24×)	339.28	4.64	195.50	20,356 (2.24×)	40.86 (1.04×)	55.05	91.83
VisionZip	CVPR'25	15%	537.27 (2.25×)	351.89	52.49	**132.72**	20,444 (2.25×)	40.09 (1.02×)	57.79	96.40
SparseVLM	ICML'25	15%	511.39 (2.37×)	341.79	**4.61**	164.80	21,479 (2.37×)	40.10 (1.02×)	52.44	87.47
FlashVID	ICLR'26 Oral	15%	583.54 (2.07×)	352.36	89.82	141.18	18,823 (2.07×)	40.92 (1.04×)	58.62	97.78
Saevia	Ours	15%	**505.60** (2.39×)	**338.62**	32.01	134.81	**21,725** (2.39×)	**41.01** (1.04×)	**58.82**	**98.12**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	15%	2191.45 (2.49×)	**1430.89**	**24.84**	735.53	14,792 (2.49×)	25.44 (1.23×)	64.89	94.00
VisionZip	CVPR'25	15%	2196.27 (2.48×)	1592.98	169.60	**433.51**	14,759 (2.48×)	**25.96** (1.26×)	66.44	96.25
SparseVLM	ICML'25	15%	2067.42 (2.64×)	1437.23	26.42	603.56	15,679 (2.64×)	25.70 (1.24×)	65.16	94.40
FlashVID	ICLR'26 Oral	15%	2181.88 (2.50×)	1603.39	122.33	455.98	14,857 (2.50×)	25.40 (1.23×)	66.15	95.82
Saevia	Ours	15%	**2020.07** (2.70×)	1439.24	120.61	460.06	**16,046** (2.70×)	25.65 (1.24×)	**66.90**	**96.91**

Table 13: Efficiency results at **20% visual token retention**.

Method		Ret. Ratio	TTFT (ms) ↓ Overall	Vision Enc.	Token Proc.	LLM Prefill	Throughput (tokens/s) ↑ Prefill	End-to-End	Accuracy Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	20%	313.16 (2.03×)	117.64	62.94	132.44	20,325 (2.03×)	39.97 (1.02×)	55.06	95.60
VisionZip	CVPR'25	20%	322.63 (1.97×)	125.80	96.66	**100.05**	19,729 (1.97×)	39.50 (1.01×)	56.83	98.67
SparseVLM	ICML'25	20%	305.44 (2.08×)	117.55	62.67	125.07	20,839 (2.08×)	40.45 (1.04×)	54.64	94.86
FlashVID	ICLR'26 Oral	20%	359.32 (1.77×)	124.83	127.05	107.25	17,714 (1.77×)	40.01 (1.02×)	57.48	99.80
Saevia	Ours	20%	**240.33** (2.64×)	**117.09**	**21.07**	102.05	**26,485** (2.64×)	**40.49** (1.04×)	**57.96**	**100.63**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	20%	568.25 (2.13×)	**337.02**	**4.54**	226.52	19,330 (2.13×)	40.43 (1.02×)	55.93	93.29
VisionZip	CVPR'25	20%	573.54 (2.11×)	351.46	53.06	**168.85**	19,151 (2.11×)	40.05 (1.01×)	58.69	97.90
SparseVLM	ICML'25	20%	**541.82** (2.23×)	340.36	4.59	196.69	**20,272** (2.23×)	40.06 (1.02×)	55.47	92.53
FlashVID	ICLR'26 Oral	20%	639.75 (1.89×)	354.30	96.43	188.84	17,169 (1.89×)	40.12 (1.02×)	58.79	98.07
Saevia	Ours	20%	544.18 (2.22×)	339.01	32.14	172.87	20,184 (2.22×)	**40.94** (1.04×)	**59.29**	**98.90**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	20%	2329.30 (2.34×)	1434.61	**23.48**	871.00	13,916 (2.34×)	25.48 (1.23×)	66.18	95.87
VisionZip	CVPR'25	20%	2309.95 (2.36×)	1595.19	133.77	**580.83**	14,033 (2.36×)	25.94 (1.26×)	67.23	97.39
SparseVLM	ICML'25	20%	2192.07 (2.49×)	**1426.98**	**23.48**	741.42	14,787 (2.49×)	25.60 (1.24×)	66.35	96.11
FlashVID	ICLR'26 Oral	20%	2339.30 (2.33×)	1603.33	127.52	608.26	13,857 (2.33×)	25.56 (1.24×)	66.80	96.76
Saevia	Ours	20%	**2179.81** (2.50×)	1434.57	121.57	623.51	**14,871** (2.50×)	**26.06** (1.26×)	**67.27**	**97.45**

Table 14: Efficiency results at **25% visual token retention**.

Method		Ret. Ratio	TTFT (ms) ↓				Throughput (tokens/s) ↑		Accuracy	
			Overall	Vision Enc.	Token Proc.	LLM Prefill	Prefill	End-to-End	Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	25%	330.15 (1.92×)	**117.47**	62.82	149.73	19,279 (1.92×)	**40.42** (1.04×)	55.62	96.57
VisionZip	CVPR'25	25%	336.84 (1.88×)	125.37	92.71	**118.63**	18,896 (1.88×)	39.94 (1.02×)	57.38	99.63
SparseVLM	ICML'25	25%	323.80 (1.96×)	118.02	63.20	142.45	19,657 (1.96×)	39.11 (1.00×)	55.97	97.18
FlashVID	ICLR'26 Oral	25%	387.71 (1.64×)	125.34	131.10	131.14	16,417 (1.64×)	40.27 (1.03×)	57.80	100.35
Saevia	Ours	25%	**263.61** (2.41×)	118.05	**21.24**	124.19	**24,146** (2.41×)	39.99 (1.02×)	**58.22**	**101.08**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	25%	608.93 (1.99×)	338.26	**4.50**	266.00	18,038 (1.99×)	40.41 (1.02×)	56.94	94.98
VisionZip	CVPR'25	25%	610.22 (1.98×)	350.04	51.04	**208.98**	18,000 (1.98×)	**40.77** (1.03×)	59.09	98.57
SparseVLM	ICML'25	25%	**580.38** (2.08×)	338.99	4.65	236.57	**18,926** (2.08×)	39.73 (1.01×)	56.92	94.95
FlashVID	ICLR'26 Oral	25%	675.03 (1.79×)	351.29	99.64	223.92	16,272 (1.79×)	40.11 (1.02×)	59.02	98.45
Saevia	Ours	25%	585.72 (2.07×)	**336.50**	32.70	216.37	18,753 (2.07×)	40.30 (1.02×)	**59.71**	**99.60**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	25%	2546.73 (2.14×)	1447.73	29.32	1069.44	12,734 (2.14×)	24.51 (1.19×)	67.12	97.23
VisionZip	CVPR'25	25%	2460.77 (2.21×)	1602.05	112.36	**746.18**	13,173 (2.21×)	25.74 (1.25×)	**68.03**	**98.55**
SparseVLM	ICML'25	25%	**2342.48** (2.33×)	**1430.90**	**29.30**	882.09	**13,838** (2.33×)	25.45 (1.23×)	67.49	97.77
FlashVID	ICLR'26 Oral	25%	2475.93 (2.20×)	1593.07	135.94	746.74	13,092 (2.20×)	**25.95** (1.26×)	66.91	96.93
Saevia	Ours	25%	2370.04 (2.30×)	1438.59	121.19	810.08	13,677 (2.30×)	25.52 (1.24×)	67.38	97.61

Table 15: Efficiency results at **30% visual token retention**.

Method		Ret. Ratio	TTFT (ms) ↓				Throughput (tokens/s) ↑		Accuracy	
			Overall	Vision Enc.	Token Proc.	LLM Prefill	Prefill	End-to-End	Avg.	Pct. (%)
LLaVA-OneVision-7B: Input 6K, Output 1K										
Full-token baseline		100%	634.26	117.43	62.86	453.83	10,035	39.04	57.60	100.00
FastV	ECCV'24	30%	352.62 (1.80×)	**116.31**	62.85	173.33	18,051 (1.80×)	39.96 (1.02×)	56.24	97.65
VisionZip	CVPR'25	30%	362.59 (1.75×)	125.21	92.35	**144.90**	17,554 (1.75×)	39.41 (1.01×)	57.54	99.90
SparseVLM	ICML'25	30%	345.47 (1.84×)	117.72	62.98	164.63	18,424 (1.84×)	39.33 (1.01×)	56.40	97.93
FlashVID	ICLR'26 Oral	30%	411.45 (1.54×)	126.05	134.81	150.46	15,470 (1.54×)	39.32 (1.01×)	57.84	100.43
Saevia	Ours	30%	**291.19** (2.18×)	119.08	**21.20**	150.78	**21,859** (2.18×)	**40.43** (1.04×)	**58.07**	**100.82**
LLaVA-Video-7B: Input 11K, Output 1K										
Full-token baseline		100%	1210.02	337.25	4.53	868.05	9,078	39.46	59.95	100.00
FastV	ECCV'24	30%	652.14 (1.86×)	339.66	4.61	307.67	16,843 (1.86×)	40.45 (1.03×)	57.44	95.81
VisionZip	CVPR'25	30%	653.70 (1.85×)	349.93	51.12	**252.48**	16,803 (1.85×)	40.89 (1.04×)	59.45	99.17
SparseVLM	ICML'25	30%	**614.50** (1.97×)	**337.30**	**4.51**	272.51	**17,875** (1.97×)	**41.35** (1.05×)	57.82	96.45
FlashVID	ICLR'26 Oral	30%	714.24 (1.69×)	351.38	100.23	262.45	15,379 (1.69×)	39.79 (1.01×)	59.33	98.97
Saevia	Ours	30%	637.19 (1.90×)	340.57	32.43	264.03	17,238 (1.90×)	40.14 (1.02×)	**59.51**	**99.27**
Qwen3-VL-8B-Instruct: Input 32K, Output 1K										
Full-token baseline		100%	5449.12	1451.10	28.34	3969.42	5,949	20.66	69.03	100.00
FastV	ECCV'24	30%	2640.07 (2.06×)	1442.81	**24.95**	1172.12	12,278 (2.06×)	24.70 (1.20×)	67.44	97.70
VisionZip	CVPR'25	30%	2618.85 (2.08×)	1611.45	96.89	**910.32**	12,378 (2.08×)	25.15 (1.22×)	**68.27**	**98.90**
SparseVLM	ICML'25	30%	**2497.40** (2.18×)	1438.63	27.88	1030.67	**12,980** (2.18×)	24.97 (1.21×)	67.94	98.43
FlashVID	ICLR'26 Oral	30%	2671.49 (2.04×)	1596.57	144.17	930.58	12,134 (2.04×)	**25.40** (1.23×)	67.33	97.54
Saevia	Ours	30%	2547.62 (2.14×)	**1435.34**	121.35	990.78	12,724 (2.14×)	25.23 (1.22×)	67.40	97.64

E.4 Detailed Component Ablation Results

Table 16 reports the benchmark-level results for the component ablations in Table 4.

Table 16: **Component ablation study across three models.** Spatial pruning is replaced with *attention-based top-K* or *random* selection under the same token budget. For each model and metric, the **best** and second-best results among all variants are bold and underlined, respectively.

Method	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-OneVision-7B								
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
w/o Temporal Merging	15%	54.07	56.44	58.58	52.88	59.50	56.29	97.73
w/o Spatial Pruning *(Attn Top-K)*	15%	**55.97**	57.74	59.91	54.23	60.39	57.65	100.09
w/o Spatial Pruning *(Random)*	15%	54.77	56.70	59.45	54.08	59.51	56.90	98.78
w/o Attention Compensation	15%	55.55	**58.19**	58.60	54.30	60.28	57.38	99.62
Saevia	15%	55.60	58.11	**60.03**	**54.53**	**61.41**	**57.94**	**100.59**
LLaVA-Video-7B								
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
w/o Temporal Merging	15%	**56.37**	61.07	56.27	53.63	63.33	58.13	96.96
w/o Spatial Pruning *(Attn Top-K)*	15%	55.70	61.15	56.67	**55.95**	64.16	58.73	97.96
w/o Spatial Pruning *(Random)*	15%	54.85	59.19	56.41	55.42	61.91	57.56	96.01
w/o Attention Compensation	15%	56.10	**62.19**	55.22	54.23	63.42	58.23	97.13
Saevia	15%	55.50	61.89	**56.75**	55.42	**64.53**	**58.82**	**98.12**
Qwen3-VL-8B-Instruct								
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
w/o Temporal Merging	15%	64.55	66.56	71.20	61.56	67.39	66.25	95.97
w/o Spatial Pruning *(Attn Top-K)*	15%	**64.82**	67.00	**71.34**	62.45	68.37	66.80	96.77
w/o Spatial Pruning *(Random)*	15%	64.28	64.41	69.49	62.60	67.81	65.72	95.20
w/o Attention Compensation	15%	64.20	**67.04**	71.18	62.98	68.15	66.71	96.64
Saevia	15%	64.80	**67.04**	71.18	**63.05**	**68.44**	**66.90**	**96.91**

E.5 Detailed Hyperparameter Results

Tables 17–20 give the numerical results for all three models in Figure 3, with 128 uniformly sampled frames for Qwen3-VL. Each sweep varies one setting from C1: the similarity threshold, maximum span, weighting endpoints, or weighting schedule. The C1 base configuration applies to these sensitivity experiments, independently of Qwen3-VL's C2 main results.

Thresholds of 0.7 and 0.8 generally yield similar averages, while 0.9 is less reliable at low token budgets. A maximum span of 32 can reduce accuracy relative to 8 or 16. Changing the tail weight or schedule produces smaller differences, and no single setting is best for all three models at every retention ratio. These observations apply to the tested ranges rather than implying insensitivity to arbitrary hyperparameter choices.

Table 17: Accuracy sensitivity to the temporal similarity threshold τ.

Setting	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-OneVision-7B								
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
$\tau = 0.70$	30%	54.77	58.67	**59.45**	**55.65**	61.51	58.01	100.71
$\tau = 0.80$	30%	**55.37**	**58.89**	59.07	55.50	61.50	**58.07**	**100.82**
$\tau = 0.90$	30%	55.35	58.56	58.87	54.08	**61.78**	57.73	100.23
$\tau = 0.70$	25%	**55.63**	58.44	59.41	**56.17**	**62.18**	**58.37**	**101.34**
$\tau = 0.80$	25%	55.50	58.96	**59.75**	54.90	62.01	58.22	101.08
$\tau = 0.90$	25%	55.25	**59.15**	59.21	54.67	62.04	58.06	100.80
$\tau = 0.70$	20%	**55.67**	58.37	59.85	**55.27**	61.51	**58.13**	**100.92**
$\tau = 0.80$	20%	**55.67**	58.15	**59.87**	54.00	**62.09**	57.96	100.63
$\tau = 0.90$	20%	55.40	**58.67**	58.99	54.08	61.62	57.75	100.26
$\tau = 0.70$	15%	55.15	**58.48**	59.91	53.63	60.69	57.57	99.95
$\tau = 0.80$	15%	**55.60**	58.11	**60.03**	**54.53**	**61.41**	**57.94**	**100.59**
$\tau = 0.90$	15%	55.00	57.85	59.33	53.63	59.82	57.13	99.18
$\tau = 0.70$	10%	54.73	**56.85**	**60.11**	**54.82**	59.55	**57.21**	**99.32**
$\tau = 0.80$	10%	55.05	56.37	59.87	54.23	**60.22**	57.15	99.22
$\tau = 0.90$	10%	**55.07**	55.26	58.68	52.21	59.43	56.13	97.45
LLaVA-Video-7B								
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
$\tau = 0.70$	30%	55.47	61.41	56.81	56.99	64.88	59.11	98.60
$\tau = 0.80$	30%	55.75	61.78	**56.99**	**57.59**	**65.42**	59.51	99.27
$\tau = 0.90$	30%	**56.45**	**62.22**	56.95	57.52	64.98	**59.62**	**99.45**
$\tau = 0.70$	25%	55.75	61.44	56.83	**57.82**	64.99	59.37	99.03
$\tau = 0.80$	25%	55.83	**62.48**	**57.46**	57.59	**65.20**	**59.71**	**99.60**
$\tau = 0.90$	25%	**55.85**	62.44	57.07	56.77	65.12	59.45	99.17
$\tau = 0.70$	20%	55.50	61.26	56.67	**56.92**	64.91	59.05	98.50
$\tau = 0.80$	20%	55.93	61.89	**57.13**	56.17	**65.31**	**59.29**	**98.90**
$\tau = 0.90$	20%	**56.10**	**61.93**	56.99	56.54	64.91	**59.29**	**98.90**
$\tau = 0.70$	15%	55.07	61.78	56.43	**55.72**	**65.03**	58.81	98.10
$\tau = 0.80$	15%	55.50	**61.89**	**56.75**	55.42	64.53	**58.82**	**98.12**
$\tau = 0.90$	15%	**55.65**	61.63	56.25	54.82	64.14	58.50	97.58
$\tau = 0.70$	10%	55.10	60.70	55.71	**54.97**	**62.93**	**57.88**	**96.55**
$\tau = 0.80$	10%	54.65	**60.93**	**55.85**	54.90	62.85	57.84	96.48
$\tau = 0.90$	10%	**55.30**	60.41	55.83	53.93	62.63	57.62	96.11
Qwen3-VL-8B-Instruct								
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
$\tau = 0.70$	30%	**64.30**	**67.59**	71.38	63.80	**69.67**	**67.35**	**97.56**
$\tau = 0.80$	30%	63.75	67.33	71.34	**64.55**	69.59	67.31	97.51
$\tau = 0.90$	30%	64.12	67.19	**71.46**	**64.55**	69.19	67.30	97.49
$\tau = 0.70$	25%	63.98	67.26	71.32	63.87	**69.25**	67.14	97.25
$\tau = 0.80$	25%	64.00	67.22	71.24	63.72	69.24	67.08	97.18
$\tau = 0.90$	25%	**64.35**	**67.33**	**71.40**	**63.95**	69.24	**67.25**	**97.42**
$\tau = 0.70$	20%	64.25	67.15	**71.18**	**64.40**	68.51	**67.10**	**97.20**
$\tau = 0.80$	20%	**64.65**	**67.22**	71.06	63.95	68.04	66.98	97.03
$\tau = 0.90$	20%	64.52	66.85	71.12	63.43	**68.78**	66.94	96.97
$\tau = 0.70$	15%	63.78	**66.59**	71.08	**63.05**	67.79	66.46	96.27
$\tau = 0.80$	15%	**64.75**	66.37	71.12	62.83	**68.29**	**66.67**	**96.58**
$\tau = 0.90$	15%	64.42	66.19	**71.20**	62.53	67.50	66.37	96.14
$\tau = 0.70$	10%	63.50	65.78	**70.46**	**62.23**	**66.04**	**65.60**	**95.03**
$\tau = 0.80$	10%	63.70	**66.07**	70.30	61.48	65.60	65.43	94.78
$\tau = 0.90$	10%	**63.90**	65.41	70.34	60.66	65.13	65.09	94.29

Table 18: Accuracy sensitivity to the maximum temporal span $s_{\max}$.

Setting	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-OneVision-7B								
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
$s_{\max} = 8$	30%	55.20	58.74	58.99	54.90	61.13	57.79	100.33
$s_{\max} = 16$	30%	**55.37**	58.89	59.07	**55.50**	**61.50**	**58.07**	**100.82**
$s_{\max} = 32$	30%	54.60	**58.93**	**59.19**	55.42	60.77	57.78	100.31
$s_{\max} = 8$	25%	55.40	**59.19**	**59.85**	54.38	**62.41**	**58.25**	**101.13**
$s_{\max} = 16$	25%	55.50	58.96	59.75	**54.90**	62.01	58.22	101.08
$s_{\max} = 32$	25%	**55.63**	58.85	59.71	54.60	61.79	58.12	100.90
$s_{\max} = 8$	20%	55.20	58.30	59.53	**54.30**	61.73	57.81	100.36
$s_{\max} = 16$	20%	**55.67**	58.15	**59.87**	54.00	**62.09**	**57.96**	**100.63**
$s_{\max} = 32$	20%	55.60	**58.41**	59.81	53.40	61.90	57.82	100.38
$s_{\max} = 8$	15%	55.43	57.74	**60.25**	**54.67**	**61.48**	57.91	100.54
$s_{\max} = 16$	15%	**55.60**	**58.11**	60.03	54.53	61.41	**57.94**	**100.59**
$s_{\max} = 32$	15%	54.75	58.04	59.95	54.08	61.38	57.64	100.07
$s_{\max} = 8$	10%	54.93	56.37	59.57	**54.45**	**60.25**	57.11	99.15
$s_{\max} = 16$	10%	**55.05**	56.37	**59.87**	54.23	60.22	**57.15**	**99.22**
$s_{\max} = 32$	10%	54.70	**56.48**	59.67	54.15	60.23	57.05	99.05
LLaVA-Video-7B								
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
$s_{\max} = 8$	30%	**56.57**	**62.07**	56.97	**57.74**	65.19	**59.71**	**99.60**
$s_{\max} = 16$	30%	55.75	61.78	**56.99**	57.59	65.42	59.51	99.27
$s_{\max} = 32$	30%	55.25	61.85	56.91	56.92	**65.46**	59.28	98.88
$s_{\max} = 8$	25%	**56.23**	**62.52**	57.38	57.37	64.98	59.70	99.58
$s_{\max} = 16$	25%	55.83	62.48	**57.46**	**57.59**	**65.20**	**59.71**	**99.60**
$s_{\max} = 32$	25%	55.10	62.22	57.26	57.44	64.95	59.39	99.07
$s_{\max} = 8$	20%	**56.65**	**61.96**	**57.32**	**56.17**	**65.51**	**59.52**	**99.28**
$s_{\max} = 16$	20%	55.93	61.89	57.13	**56.17**	65.31	59.29	98.90
$s_{\max} = 32$	20%	55.10	61.63	56.93	56.10	64.62	58.88	98.22
$s_{\max} = 8$	15%	**56.25**	61.52	56.71	**55.87**	64.19	**58.91**	**98.27**
$s_{\max} = 16$	15%	55.50	**61.89**	**56.75**	55.42	**64.53**	58.82	98.12
$s_{\max} = 32$	15%	55.03	61.63	56.65	55.65	64.24	58.64	97.81
$s_{\max} = 8$	10%	**55.53**	60.81	**55.89**	54.67	62.57	**57.89**	**96.56**
$s_{\max} = 16$	10%	54.65	**60.93**	55.85	**54.90**	62.85	57.84	96.48
$s_{\max} = 32$	10%	54.85	60.52	55.56	54.45	**62.92**	57.66	96.18
Qwen3-VL-8B-Instruct								
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
$s_{\max} = 8$	30%	**65.02**	67.30	71.46	**64.62**	**69.78**	**67.64**	**97.98**
$s_{\max} = 16$	30%	63.75	67.33	71.34	64.55	69.59	67.31	97.51
$s_{\max} = 32$	30%	63.38	**67.48**	**71.52**	64.47	69.40	67.25	97.42
$s_{\max} = 8$	25%	**65.25**	67.37	**71.28**	**63.72**	69.17	**67.36**	**97.57**
$s_{\max} = 16$	25%	64.00	67.22	71.24	**63.72**	**69.24**	67.08	97.18
$s_{\max} = 32$	25%	63.78	**67.56**	71.02	63.13	68.95	66.89	96.89
$s_{\max} = 8$	20%	**65.22**	**67.30**	**71.14**	63.80	**68.52**	**67.20**	**97.34**
$s_{\max} = 16$	20%	64.65	67.22	71.06	**63.95**	68.04	66.98	97.03
$s_{\max} = 32$	20%	63.92	66.96	**71.14**	63.72	68.31	66.81	96.78
$s_{\max} = 8$	15%	64.58	**66.59**	**71.26**	62.68	68.21	66.66	96.57
$s_{\max} = 16$	15%	**64.75**	66.37	71.12	**62.83**	**68.29**	**66.67**	**96.58**
$s_{\max} = 32$	15%	63.65	66.22	71.16	**62.83**	67.90	66.35	96.12
$s_{\max} = 8$	10%	63.65	65.89	70.24	61.18	65.51	65.29	94.59
$s_{\max} = 16$	10%	**63.70**	**66.07**	**70.30**	**61.48**	**65.60**	**65.43**	**94.78**
$s_{\max} = 32$	10%	62.52	65.78	70.22	60.51	65.53	64.91	94.03

Table 19: Accuracy sensitivity to the request-aware weighting endpoints.

$(\alpha_{\text{head}}, \alpha_{\text{tail}})$	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-OneVision-7B								
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
(0.95, 0.10)	30%	55.10	58.81	**59.75**	55.05	61.73	58.09	100.85
(0.95, 0.30)	30%	55.20	**59.07**	59.55	55.05	**61.97**	**58.17**	**100.99**
(0.95, 0.50)	30%	**55.37**	58.89	59.07	**55.50**	61.50	58.07	100.82
(0.95, 0.10)	25%	**55.55**	58.07	59.69	54.75	**62.63**	58.14	100.94
(0.95, 0.30)	25%	55.27	58.78	59.63	**55.12**	62.30	**58.22**	**101.08**
(0.95, 0.50)	25%	55.50	**58.96**	**59.75**	54.90	62.01	**58.22**	**101.08**
(0.95, 0.10)	20%	**55.75**	58.30	**59.91**	54.60	61.71	58.05	100.79
(0.95, 0.30)	20%	55.70	**58.37**	59.81	**54.90**	61.62	**58.08**	**100.84**
(0.95, 0.50)	20%	55.67	58.15	59.87	54.00	**62.09**	57.96	100.63
(0.95, 0.10)	15%	55.17	57.19	59.61	**55.20**	61.19	57.67	100.13
(0.95, 0.30)	15%	55.50	57.74	**60.05**	54.75	60.99	57.81	100.36
(0.95, 0.50)	15%	**55.60**	**58.11**	60.03	54.53	**61.41**	**57.94**	**100.59**
(0.95, 0.10)	10%	**55.07**	55.89	59.81	53.70	58.98	56.69	98.43
(0.95, 0.30)	10%	54.73	**56.70**	59.45	**55.42**	59.88	**57.24**	**99.38**
(0.95, 0.50)	10%	55.05	56.37	**59.87**	54.23	**60.22**	57.15	99.22
LLaVA-Video-7B								
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
(0.95, 0.10)	30%	55.80	61.81	56.91	57.22	**65.67**	59.48	99.23
(0.95, 0.30)	30%	**55.83**	**62.15**	56.91	**58.04**	65.51	**59.69**	**99.57**
(0.95, 0.50)	30%	55.75	61.78	**56.99**	57.59	65.42	59.51	99.27
(0.95, 0.10)	25%	55.35	61.85	57.25	**58.04**	65.14	59.53	99.30
(0.95, 0.30)	25%	55.70	61.85	57.19	57.22	**65.20**	59.43	99.14
(0.95, 0.50)	25%	**55.83**	**62.48**	**57.46**	57.59	**65.20**	**59.71**	**99.60**
(0.95, 0.10)	20%	55.60	62.33	57.07	**57.22**	64.62	**59.37**	**99.03**
(0.95, 0.30)	20%	55.63	**62.41**	57.05	56.92	64.42	59.29	98.90
(0.95, 0.50)	20%	**55.93**	61.89	**57.13**	56.17	**65.31**	59.29	98.90
(0.95, 0.10)	15%	**55.73**	**62.00**	56.83	55.65	63.56	58.75	98.01
(0.95, 0.30)	15%	55.33	61.74	**57.15**	**55.72**	64.15	**58.82**	**98.12**
(0.95, 0.50)	15%	55.50	61.89	56.75	55.42	**64.53**	**58.82**	**98.12**
(0.95, 0.10)	10%	**55.00**	60.56	**56.21**	**55.42**	**63.11**	**58.06**	**96.85**
(0.95, 0.30)	10%	54.97	60.74	55.81	54.97	62.95	57.89	96.57
(0.95, 0.50)	10%	54.65	**60.93**	55.85	54.90	62.85	57.84	96.48
Qwen3-VL-8B-Instruct								
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
(0.95, 0.10)	30%	**64.50**	**67.41**	**71.36**	63.43	**70.11**	**67.36**	**97.58**
(0.95, 0.30)	30%	64.08	67.07	71.16	63.35	69.58	67.05	97.13
(0.95, 0.50)	30%	63.75	67.33	71.34	**64.55**	69.59	67.31	97.51
(0.95, 0.10)	25%	**64.15**	67.11	**71.28**	63.28	68.93	66.95	96.99
(0.95, 0.30)	25%	64.08	**67.37**	71.14	63.43	68.71	66.95	96.98
(0.95, 0.50)	25%	64.00	67.22	71.24	**63.72**	**69.24**	**67.08**	**97.18**
(0.95, 0.10)	20%	64.28	66.74	70.98	**64.10**	68.32	66.88	96.89
(0.95, 0.30)	20%	**64.70**	67.11	71.02	63.65	**68.84**	**67.06**	**97.15**
(0.95, 0.50)	20%	64.65	**67.22**	**71.06**	63.95	68.04	66.98	97.03
(0.95, 0.10)	15%	63.98	66.96	**71.20**	62.53	68.22	66.58	96.45
(0.95, 0.30)	15%	64.32	**67.00**	71.10	**63.20**	68.16	**66.76**	**96.71**
(0.95, 0.50)	15%	**64.75**	66.37	71.12	62.83	**68.29**	66.67	96.58
(0.95, 0.10)	10%	63.42	**66.15**	70.15	61.48	65.79	65.40	94.74
(0.95, 0.30)	10%	63.15	66.07	**70.32**	**61.63**	**65.82**	65.40	94.74
(0.95, 0.50)	10%	**63.70**	66.07	70.30	61.48	65.60	**65.43**	**94.78**

Table 20: Accuracy sensitivity to the request-aware weighting schedule.

Setting	Ret. Ratio	MVBench	Video-MME	EgoSchema$_{full}$	LVB$_{val.}$	MLVU$^{Dev}_{M-Avg}$	Avg.	Pct. (%)
LLaVA-OneVision-7B								
Full-token baseline	100%	55.15	58.33	58.58	54.23	61.69	57.60	100.00
Linear	30%	55.03	58.81	59.25	54.45	61.39	57.79	100.33
Convex log	30%	55.33	58.74	**59.27**	54.82	**61.95**	58.02	100.74
Concave log	30%	**55.37**	**58.89**	59.07	**55.50**	61.50	**58.07**	**100.82**
Linear	25%	**55.80**	59.00	**59.75**	55.72	61.76	58.41	101.41
Convex log	25%	55.30	**59.26**	59.63	**56.17**	**62.02**	**58.48**	**101.53**
Concave log	25%	55.50	58.96	**59.75**	54.90	62.01	58.22	101.08
Linear	20%	55.63	58.30	59.73	54.60	61.68	57.99	100.68
Convex log	20%	**56.03**	**58.48**	59.73	**55.12**	61.16	**58.10**	**100.88**
Concave log	20%	55.67	58.15	**59.87**	54.00	**62.09**	57.96	100.63
Linear	15%	**55.85**	57.93	**60.05**	54.67	60.52	57.80	100.35
Convex log	15%	55.77	57.93	59.97	**54.75**	60.17	57.72	100.21
Concave log	15%	55.60	**58.11**	60.03	54.53	**61.41**	**57.94**	**100.59**
Linear	10%	**55.35**	**56.44**	**59.99**	54.53	60.17	57.30	99.48
Convex log	10%	55.30	56.41	59.95	**55.05**	60.18	**57.38**	**99.62**
Concave log	10%	55.05	56.37	59.87	54.23	**60.22**	57.15	99.22
LLaVA-Video-7B								
Full-token baseline	100%	57.23	63.85	56.15	56.54	65.96	59.95	100.00
Linear	30%	55.83	**62.26**	56.69	57.37	65.32	59.49	99.23
Convex log	30%	**55.93**	62.11	56.75	57.22	**65.63**	**59.53**	**99.30**
Concave log	30%	55.75	61.78	**56.99**	**57.59**	65.42	59.51	99.27
Linear	25%	**56.13**	62.04	57.19	57.07	65.43	59.57	99.37
Convex log	25%	55.87	62.44	57.28	56.54	**65.55**	59.54	99.32
Concave log	25%	55.83	**62.48**	**57.46**	**57.59**	65.20	**59.71**	**99.60**
Linear	20%	55.65	**62.48**	56.79	56.25	64.85	59.20	98.75
Convex log	20%	55.83	62.07	56.77	**56.62**	64.92	59.24	98.82
Concave log	20%	**55.93**	61.89	**57.13**	56.17	**65.31**	**59.29**	**98.90**
Linear	15%	55.75	61.37	**56.89**	**56.32**	63.96	**58.86**	**98.18**
Convex log	15%	**55.93**	60.89	56.69	55.95	63.78	58.65	97.83
Concave log	15%	55.50	**61.89**	56.75	55.42	**64.53**	58.82	98.12
Linear	10%	54.90	60.67	**56.07**	54.97	62.48	57.82	96.45
Convex log	10%	**55.17**	60.74	55.97	**55.12**	62.33	**57.87**	**96.53**
Concave log	10%	54.65	**60.93**	55.85	54.90	**62.85**	57.84	96.48
Qwen3-VL-8B-Instruct								
Full-token baseline	100%	66.83	69.11	72.55	63.95	72.72	69.03	100.00
Linear	30%	64.40	67.30	71.26	63.80	69.45	67.24	97.41
Convex log	30%	**64.42**	67.26	71.16	64.47	**69.63**	**67.39**	**97.62**
Concave log	30%	63.75	**67.33**	**71.34**	**64.55**	69.59	67.31	97.51
Linear	25%	**64.72**	**67.30**	71.46	**63.72**	69.02	**67.24**	**97.41**
Convex log	25%	64.50	**67.30**	**71.52**	63.65	69.03	67.20	97.35
Concave log	25%	64.00	67.22	71.24	**63.72**	**69.24**	67.08	97.18
Linear	20%	64.10	**67.30**	71.14	64.17	**68.73**	**67.09**	97.18
Convex log	20%	64.32	67.22	**71.18**	**64.32**	68.42	**67.09**	**97.19**
Concave log	20%	**64.65**	67.22	71.06	63.95	68.04	66.98	97.03
Linear	15%	64.35	66.48	**71.18**	62.38	**68.29**	66.54	96.38
Convex log	15%	64.30	**66.93**	71.04	62.15	68.25	66.53	96.38
Concave log	15%	**64.75**	66.37	71.12	**62.83**	**68.29**	**66.67**	**96.58**
Linear	10%	**63.78**	66.00	70.26	**61.48**	**66.19**	**65.54**	**94.95**
Convex log	10%	**63.78**	**66.15**	**70.30**	61.03	65.92	65.44	94.79
Concave log	10%	63.70	66.07	**70.30**	**61.48**	65.60	65.43	94.78

F Additional Case Studies

Case study. Figure 5 shows Saevia correctly identifying two pandas on the swing in a Video-MME example. Merged tokens occupy their units' first positions, so temporal merging removes no tokens from the first frame. Overall, spatial pruning accounts for most of the token reduction, while temporal merging removes redundant tokens across frames.

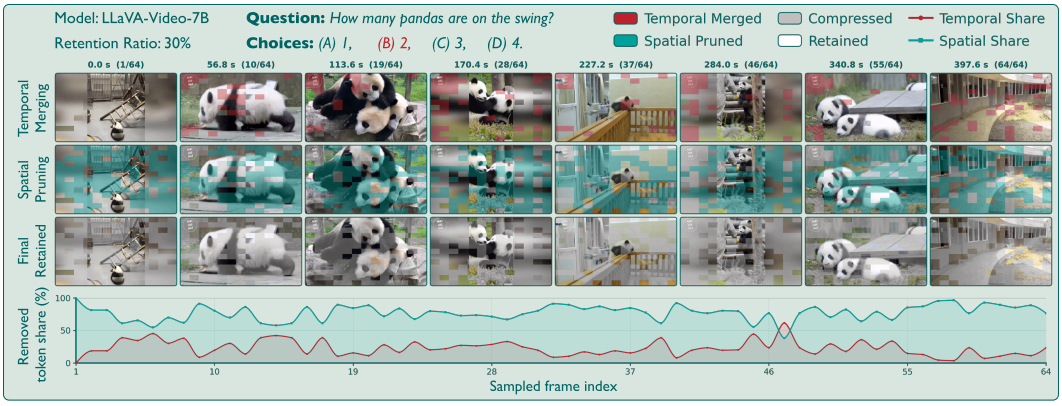

Figure 5: **Case study.** A Video-MME example with 8 of 64 uniformly sampled frames. From top to bottom: tokens removed by temporal merging, removed by spatial pruning, and final retained tokens. The bottom plot shows each stage's share of removed tokens per frame.

Figures 6–10 illustrate five tasks from the evaluated benchmarks, using the same C1 configuration as the LLaVA main experiments. The examples cover action counting, question-specific evidence, egocentric activity, temporal order, and anomaly recognition at target retention ratios of 10%, 20%, or 30%. They illustrate token selection in individual videos and do not establish benchmark-wide accuracy or localization performance.

Each figure shows eight uniformly spaced frames from the sampled sequence. The three rows mark tokens removed by temporal merging, tokens removed by spatial pruning, and retained tokens. The bottom plot shows how each stage contributes to the removed-token count within each frame. These plots count visual patches; model-specific separator tokens are excluded, so the patch-only retained fraction can differ slightly from the target input-token retention ratio.

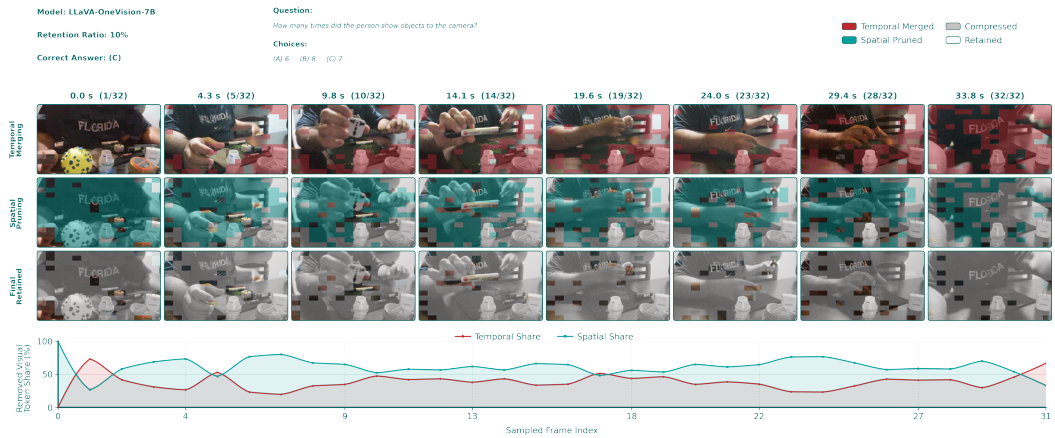

Figure 6: **Action counting on MVBench.** Saevia uses LLaVA-OneVision-7B with C1 and a 10% target retention ratio. Eight of the 32 sampled frames are shown. The rows identify tokens removed by temporal merging, removed by spatial pruning, and retained, respectively. The bottom plot reports the per-frame removal shares.

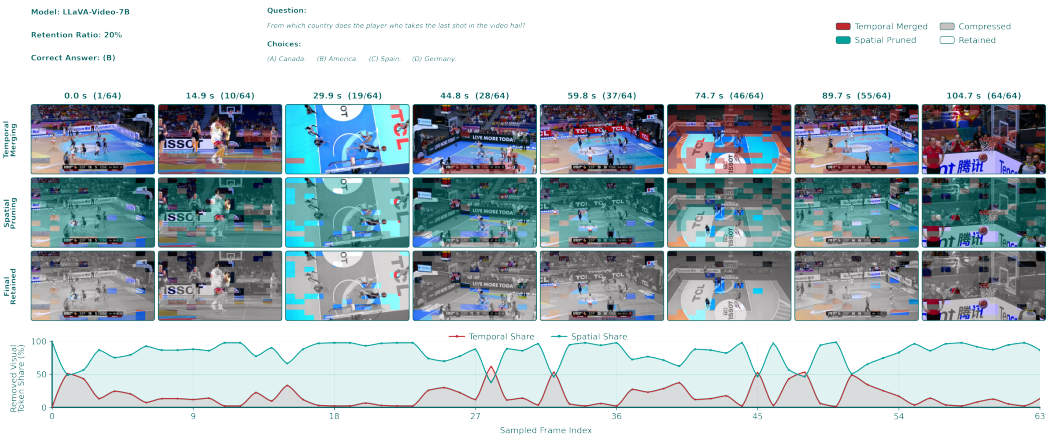

Figure 7: **Question-specific evidence on Video-MME.** Saevia uses LLaVA-Video-7B with C1 and a 20% target retention ratio. Eight of the 64 sampled frames are shown. The question concerns the player taking the final shot. The rows and bottom plot follow the token-removal conventions in Figure 6.

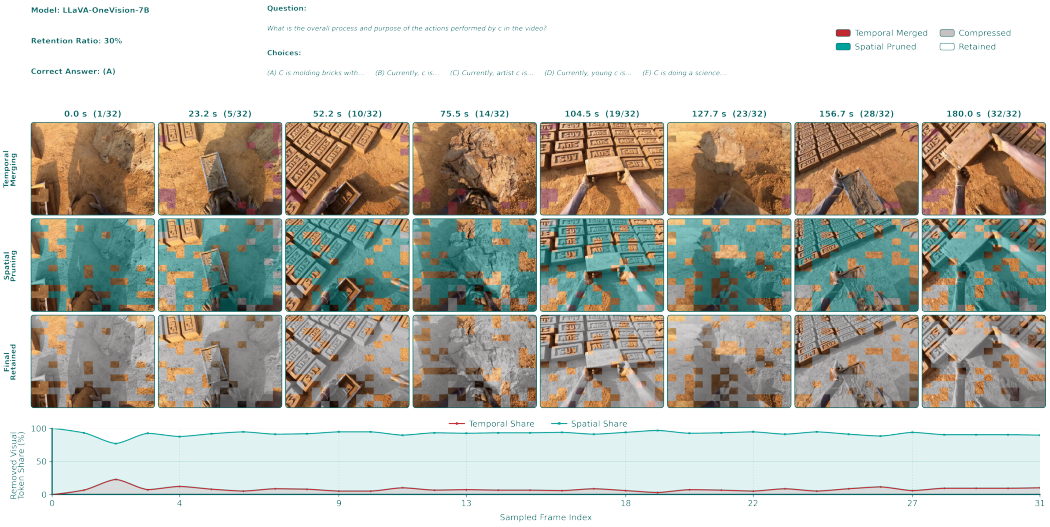

Figure 8: **Egocentric activity on EgoSchema.** Saevia uses LLaVA-OneVision-7B with C1 and a 30% target retention ratio. Eight of the 32 sampled frames are shown. The question asks about the overall process and purpose of the activity. The rows and bottom plot follow Figure 6.

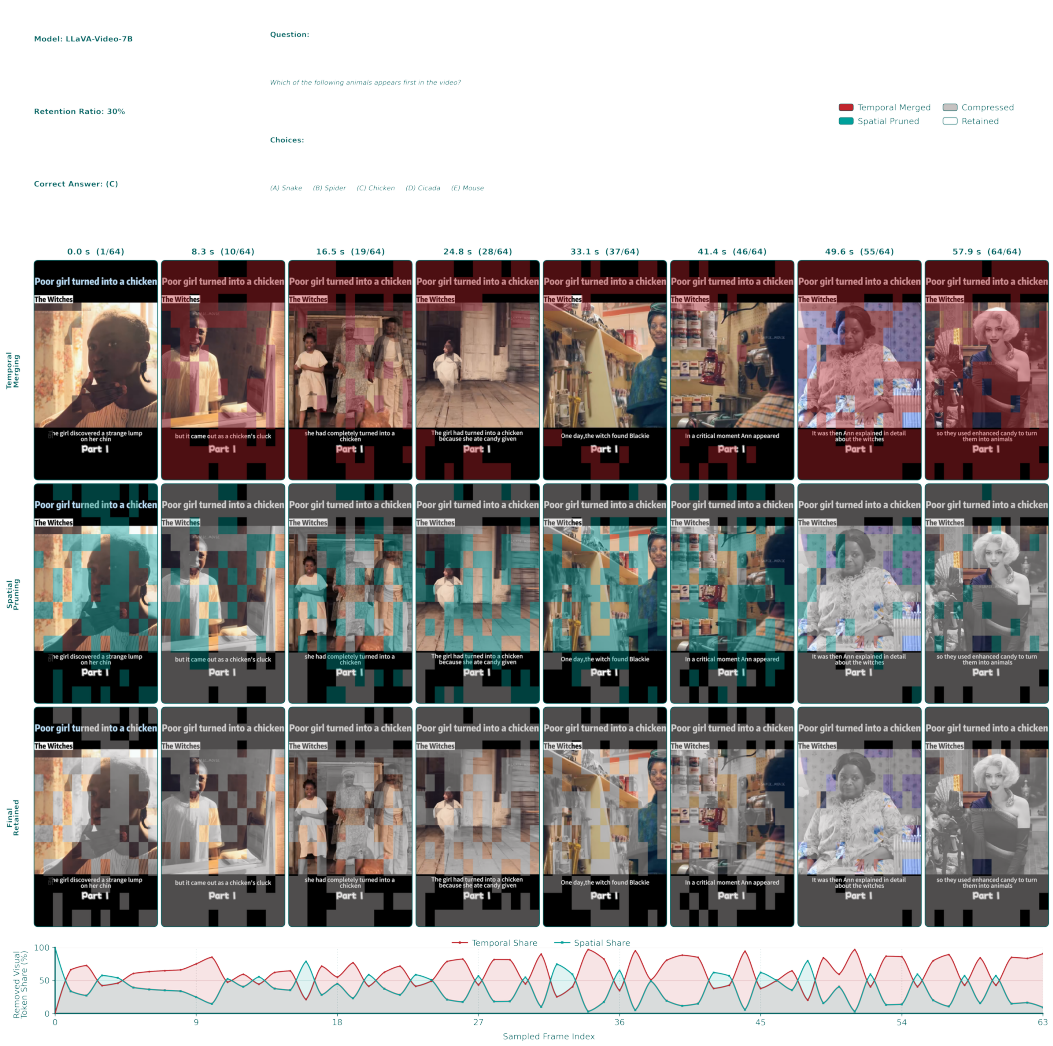

Figure 9: **Temporal order on LongVideoBench.** Saevia uses LLaVA-Video-7B with C1 and a 30% target retention ratio. Eight of the 64 sampled frames are shown. The question asks which animal appears first. The rows and bottom plot follow Figure 6.

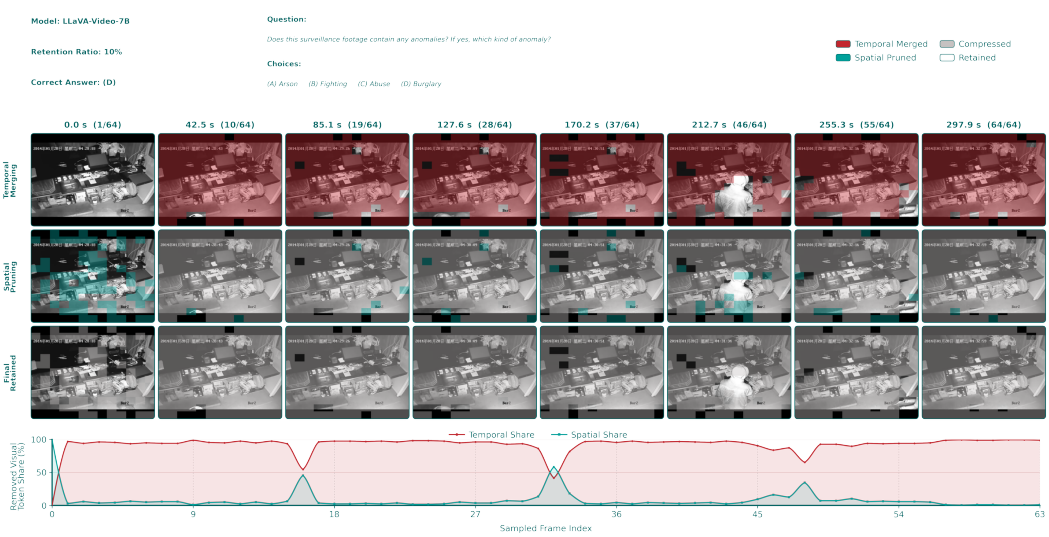

Figure 10: **Anomaly recognition on MLVU.** Saevia uses LLaVA-Video-7B with C1 and a 10% target retention ratio. Eight of the 64 sampled frames are shown. The question asks whether the surveillance video contains an anomaly and, if so, its type. The rows and bottom plot follow Figure 6.