DYNVID: ADAPTIVE DYNAMISM-AWARE TOKEN COMPRESSION FOR EFFICIENT VIDEO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Large Language Models (VLLMs) have achieved remarkable success in video understanding, yet suffer from prohibitive computational cost due to the massive number of visual tokens from densely sampled frames. While visual token compression offers a promising solution, existing methods are limited in two respects: they either allocate a uniform token budget across all frames, neglecting the inherent variation in scene complexity, or primarily rely on attention scores for token selection, overlooking temporal dynamics unique to video. To address these limitations, we propose **DynVid**, a training-free, dynamism-aware adaptive token compression framework for VLLMs that follows an *allocate-then-select* paradigm. In allocation stage, video frames are partitioned into semantically coherent segments via dynamic segmentation, and each segment receives an adaptive token budget proportional to its visual complexity. In selection stage, we introduce a token-level dynamism score as a temporally-aware importance signal that complements attention scores, and employ a Facility Location objective to select anchor tokens that are both individually important and collectively representative. Extensive experiments on eight video understanding benchmarks demonstrate that DynVid outperforms training-free baselines in both accuracy and inference efficiency. Notably, on LLaVA-OneVision, DynVid retains **99.3%** relative accuracy with **10%** average retention ratio of visual tokens preserved, and still maintains **96.4%** relative accuracy at an aggressive **5%** retention ratio. On Qwen2.5-VL, DynVid achieves **97.4%** and **96.0%** relative accuracy at 10% and 7.5% retention ratios, respectively, while delivering up to **7.53**$\times$ and **9.49**$\times$ prefilling speedups.

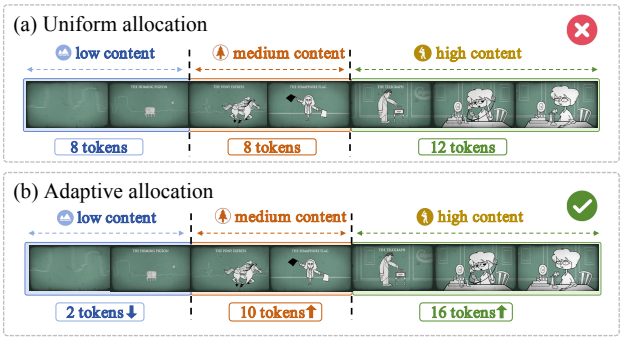
1 INTRODUCTION

The convergence of large language models (LLMs) (OpenAI, 2023; Llama Team, 2024) and visual encoders (Dosovitskiy et al., 2021; Radford et al., 2021) has given rise to Video Large Language Models (VLLMs) (Liu et al., 2023; Li et al., 2024a; Zhang et al., 2024; Wang et al., 2024; Qwen Team, 2025b; Chen et al., 2024c; Li et al., 2025) that achieve strong performance on video understanding tasks. These models encode each sampled frame into hundreds of visual tokens via a vision encoder (Dosovitskiy et al., 2021; Zhai et al., 2023), easily accumulating tens of thousands in total even for moderately sampled videos. Given the quadratic complexity of the LLM's self-attention, this incurs substantial computational cost and memory consumption during inference.

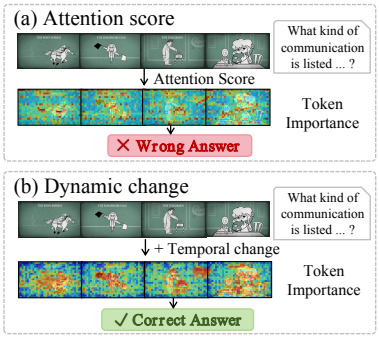
Fortunately, visual tokens in video contain significant spatial and temporal redundancy, motivating extensive work on token compression. Early methods focus on image understanding: one line prunes less important tokens based on attention scores from the LLM decoder or visual encoder (Chen et al., 2024a; Zhang et al., 2025b; Yang et al., 2025; Xing et al., 2025), while another merges similar tokens into compact representations (Bolya et al., 2023; Yang et al., 2025). Recent approaches extend these ideas to video by exploiting inter-frame temporal redundancy through spatiotemporal merging or pruning (Shen et al., 2025; Fan et al., 2026).

Limitations. Despite this progress, existing methods share two fundamental limitations: (1) Uniform Budget Allocation. Many existing methods allocate a uniform token budget across all frames, assuming equal information density. However, real-world videos exhibit significant complexity variation, as a static shot carries far less information than a rapid action sequence, and uniform allocation inevitably wastes budget on simple scenes while starving complex ones. Some approaches attempt

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.



(a) Uniform vs. adaptive allocation.



(b) Token importance comparison.

Figure 1: Limitations of existing methods. (a) Uniform allocation ignores content complexity; adaptive allocation distributes tokens proportionally to scene complexity. (b) Relying solely on attention scores overlooks temporal dynamics; incorporating temporal change preserves temporally informative tokens that attention alone would miss.

to mitigate this by first applying uniform per-frame allocation and then merging temporally redundant tokens across frames. Yet such post-hoc merging inevitably dilutes or discards representative tokens, leading to inefficient utilization of the token budget. In contrast, as shown in Figure 1a, an ideal strategy should proactively allocate more tokens to complex, dynamic segments while reducing the budget for static, low-content ones, thereby providing a well-calibrated foundation for subsequent anchor token selection. (2) Neglecting Video Dynamism. Existing methods determine token importance primarily through attention scores (*e.g.*, `[CLS]` attention), which measure how heavily each token is attended within a single frame. This captures *where* salient content resides but is blind to *how* that content changes over time. Consequently, tokens undergoing significant temporal feature change are systematically undervalued, while temporally redundant tokens in stable regions are over-retained, potentially discarding query-relevant information. As shown in Figure 1b, incorporating temporal change as an importance signal can better preserve dynamic content that attention scores alone would miss. Therefore, an effective video token compression method should make every retained token count by maximizing representativeness and content coverage. This requires two capabilities: (i) automatically allocating token budgets based on segment complexity, so that tokens are directed to where visual content is rich rather than wasted on redundant scenes; and (ii) under a given budget, selecting tokens that are individually important and collectively representative, covering all aspects of the visual content (*e.g.*, both static and dynamic regions).

Our Solution. To this end, we propose **DynVid** (**Dyn**amism-Aware Adaptive **Vid**eo Token Compression), a training-free framework that follows an allocate-then-select paradigm, making every retained token count. In the allocation stage, we introduce **E**ffective **R**ank-Based Adaptive Budget **A**llocation (**ERA**): after segmenting the video, each segment receives a token budget proportional to its visual complexity, measured by the effective rank of its feature matrix. This ensures that complex, content-rich segments retain more tokens while simple, redundant segments are aggressively compressed. In the selection stage, we introduce **D**ynamism-**A**ware **T**oken **S**election (**DATS**): a token-level dynamism score, composed of temporal novelty and temporal asymmetry, is combined with attention scores to form a composite token importance measure, and a Facility Location (FL) objective selects anchor tokens that maximize both importance and content coverage. This dual-signal scoring with coverage-aware selection ensures that the retained tokens are both individually informative and collectively representative of the full visual content. Extensive experiments on eight video understanding benchmarks with four representative VLLMs, including LLaVA-OneVision (Li et al., 2024a), LLaVA-Video (Zhang et al., 2024), Qwen2.5-VL (Qwen Team, 2025a), and Qwen3-VL (Qwen Team, 2025b), validate the effectiveness of our DynVid. Notably, on LLaVA-OneVision, DynVid retains 99.3% of the original accuracy with only 10% of visual tokens preserved, and still maintains 96.4% accuracy at an aggressive 5% retention ratio. On Qwen2.5-VL, DynVid achieves 97.4% and 96.0% relative accuracy at 10% and 7.5% retention ratios, respectively, while delivering up to 7.53× and 9.49× prefilling speedup.

Overall, our contributions can be summarized as follows:

- We identify two fundamental yet underexplored aspects for video token compression: (i) adaptive budget allocation based on segment complexity, and (ii) incorporating temporal dynamism into token selection to preserve broader visual content.
- We propose DynVid, a training-free framework following an *allocate-then-select* paradigm. Specifically, Effective Rank-based Adaptive Allocation (ERA) assigns token budgets proportional to scene complexity; Dynamism-Aware Token Selection (DATS) incorporates temporal dynamism as a complementary importance signal and employs a Facility Location objective to select anchor tokens that jointly maximize importance and coverage.
- Extensive experiments on eight benchmarks with four representative VLLMs demonstrate that DynVid outperforms training-free baselines in both accuracy and inference efficiency across diverse video understanding tasks.

2 Related Work

Multimodal Large Language Models. Recent advances in LLMs (Touvron et al., 2023; OpenAI, 2023) have driven the development of multimodal LLMs (MLLMs) that integrate visual perception with language reasoning (Liu et al., 2023; Dai et al., 2023; Ye et al., 2023). Extending to video tasks, representative models such as LLaVA-OneVision (Li et al., 2024a) and LLaVA-Video (Zhang et al., 2024) encode each frame independently and concatenate the resulting tokens, while Qwen2-VL (Wang et al., 2024) and Qwen2.5-VL (Qwen Team, 2025a) integrate temporal modeling directly into the visual encoder through 3D convolutions. Despite architectural differences, all approaches share a common bottleneck: the number of visual tokens grows linearly with the number of frames, creating substantial computational cost during inference and motivating efficient token compression.

Visual Token Compression. To address token redundancy, training-free compression methods have been proposed for both images and videos. For images, attention-based pruning methods (Chen et al., 2024a; Zhang et al., 2025b; Xing et al., 2025) leverage the LLM's attention to identify and remove dispensable tokens at designated layers, while merging-based methods (Bolya et al., 2023; Yang et al., 2025; Zhong et al., 2025) combine redundant tokens via similarity matching to avoid hard information loss. However, these image-centric methods treat each frame independently when applied to video, failing to exploit temporal redundancy. For video, recent works explicitly address cross-frame redundancy: PruneVid (Huang et al., 2025) merges spatiotemporal tokens and then applies text-guided pruning; FastVID (Shen et al., 2025) partitions videos into temporal segments and applies density-based selection within each segment, but distributes budgets without considering segment complexity; FlashVID (Fan et al., 2026) introduces tree-based spatiotemporal merging that jointly exploits spatial and temporal redundancy, yet its token selection relies on attention score and diversity without modeling inter-frame temporal change; and others (Fu et al., 2025b; Shao et al., 2025; Liu et al., 2025) explore cascaded compression or instruction-conditioned strategies.

Despite these advances, existing methods lack proactive adaptive allocation based on segment complexity, and their token-selection signals do not explicitly model inter-frame temporal change. DynVid addresses both gaps through effective rank-based budget allocation and dynamism-aware importance scoring. A comprehensive discussion is provided in Appendix A.

3 Preliminaries

A VLLM consists of a visual encoder $\mathcal{V}$ (*e.g.*, ViT (Dosovitskiy et al., 2021) or SigLIP (Zhai et al., 2023)), a projection module $\mathcal{P}$, and a LLM $\mathcal{M}$. Given an input video with T frames, the visual encoder processes each frame into a sequence of patch tokens. After projection, the video is represented as a token sequence $\mathbf{V} = [v_1, \ldots, v_N] \in \mathbb{R}^{N \times d}$, where $N = T \times n$, n is the number of tokens per frame, and d is the hidden dimension. These visual tokens are concatenated with text tokens and fed into the LLM for autoregressive generation. Since visual tokens typically dominate the input (*e.g.*, N can reach tens of thousands while text tokens remain in the hundreds), the self-attention cost scales as $\mathcal{O}(N^2 d)$ and KV cache memory grows as $\mathcal{O}(Nd)$, both of which become prohibitive for long videos. Visual token compression aims to reduce N tokens to $B \ll N$ while preserving the information necessary for video understanding. Our method operates before the LLM and addresses two core questions: *how many* tokens to allocate, and *which* tokens to retain within each segment.

4 METHOD

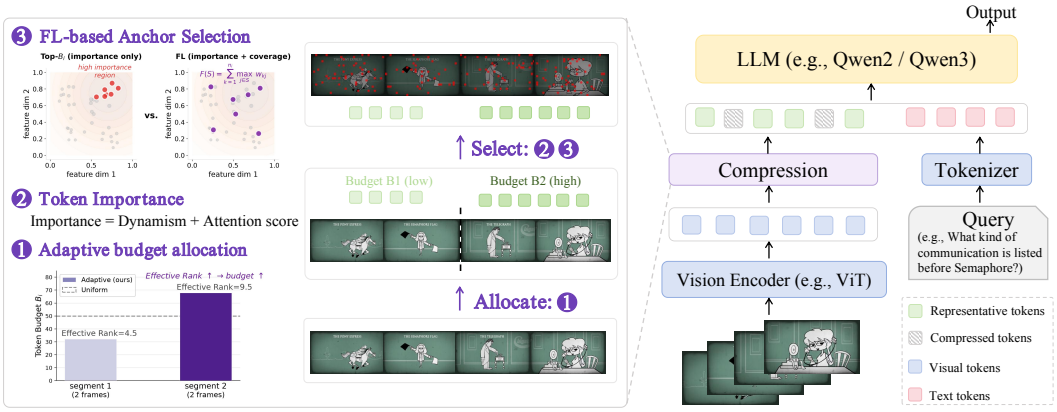


Figure 2: **Overview of DynVid.** Given a video, we first partition frames into segments and allocate per-segment token budgets based on effective rank (ERA). Then, within each segment, we select tokens via dynamism-aware importance scoring and Facility Location optimization (DATS).

To address these two questions, we present DynVid (as shown in Figure 2), a training-free token compression framework that follows an allocate-then-select paradigm. In the allocation stage, we propose Effective Rank-based Adaptive Allocation (ERA), which is motivated by entropy-based effective rank and uses the participation ratio as an efficient surrogate in practice to estimate segment-level visual complexity and assign per-segment budgets proportionally. In the selection stage, we propose Dynamism-Aware Token Selection (DATS), which identifies the most representative tokens that maximize content coverage within each segment, using dynamism-aware importance scoring combined with facility location optimization.

4.1 EFFECTIVE RANK-BASED BUDGET ALLOCATION (ERA)

Temporal Segmentation. Before allocating token budgets, we partition the T video frames into G visually coherent segments of consecutive frames. This ensures that subsequent budget allocation and token selection operate on temporally consistent units. Following prior work (Shen et al., 2025), we place segment boundaries where the normalized similarity between mean-pooled features of adjacent frames falls below a threshold τ_s.

Effective Rank-Based Allocation. Given G segments and a total token budget B, we assign each segment g_i a budget B_i ($\sum_i B_i = B$) according to its visual complexity, reflecting the insight that not all segments are equally informative: a segment with diverse, non-redundant content requires more tokens to preserve its information, whereas a highly redundant segment can be compressed more aggressively. We measure visual complexity by the *effective rank* (Roy & Vetterli, 2007), which quantifies the intrinsic dimensionality of a feature matrix through the entropy of its normalized singular value spectrum. Let segment g_i contain $n_i = T_i \times n$ visual tokens with feature matrix $\mathbf{V}_i \in \mathbb{R}^{n_i \times d}$, where T_i is the number of frames. The effective rank is defined as:

$$\text{erank}(\mathbf{V}_i) = \exp\left(-\sum_{j=1}^{r_i} \tilde{\sigma}_j \log \tilde{\sigma}_j\right), \quad \tilde{\sigma}_j = \frac{\sigma_j(\mathbf{V}_i)}{\sum_{k=1}^{r_i} \sigma_k(\mathbf{V}_i)}, \tag{1}$$

where $\sigma_j(\mathbf{V}_i)$ are the singular values and $r_i = \text{rank}(\mathbf{V}_i)$. A segment with diverse token features exhibits a flatter spectrum and higher effective rank, whereas highly redundant and spectrally concentrated features yield a lower effective rank. Motivated by this observation, we adopt a simple proportional allocation rule:

$$B_i = \left\lfloor B \cdot \frac{\text{erank}(\mathbf{V}_i)}{\sum_{g=1}^{G} \text{erank}(\mathbf{V}_g)} \right\rfloor. \tag{2}$$

We enforce $B_i \geq 1$ for every segment (with $B \geq G$ in all evaluated settings). After rounding, we adjust the integer budgets according to their allocation weights, adding tokens to higher-weight seg-

ments or removing tokens from lower-weight segments when necessary, subject to each segment's available token count, until $\sum_i B_i = B$.

The following proposition characterizes the reconstruction behavior of higher-effective-rank spectra under a power-law spectral model, providing motivation for effective-rank-guided budget allocation.

Proposition 1 (Spectral Flatness and Relative Reconstruction Error)**.** *Suppose the singular values of a feature matrix $\mathbf{V}$ follow a power-law decay $\sigma_j(\mathbf{V}) = cj^{-\alpha}$ for $j = 1, \ldots, r$, where $c > 0$ and $\alpha > 0$. For a rank-b approximation, define the relative reconstruction error as*

$$\epsilon_\alpha(b) = \frac{\sum_{j=b+1}^{r} \sigma_j^2(\mathbf{V})}{\sum_{j=1}^{r} \sigma_j^2(\mathbf{V})}, \qquad 1 \le b < r. \tag{3}$$

For fixed r and b, both $\mathrm{erank}(\mathbf{V})$ and $\epsilon_\alpha(b)$ decrease monotonically with α. Consequently, within this spectral family, a flatter spectrum with higher effective rank incurs larger relative reconstruction error under the same rank budget.

Proof. Let $p_j(\alpha) = j^{-\alpha} / \sum_{k=1}^{r} k^{-\alpha}$. The entropy $H(\alpha) = -\sum_j p_j(\alpha) \log p_j(\alpha)$ satisfies $\frac{dH}{d\alpha} = -\alpha \operatorname{Var}_{p(\alpha)}(\log j) \le 0$, so $\mathrm{erank}(\mathbf{V}) = \exp(H(\alpha))$ decreases with α. For the relative reconstruction error, the scale factor c cancels. Let $q_j = j^{-2\alpha} / \sum_{k=1}^{r} k^{-2\alpha}$ and $\epsilon_\alpha(b) = \sum_{j>b} q_j$. Then

$$\frac{d\epsilon_\alpha(b)}{d\alpha} = -2\epsilon_\alpha(b)\big(1 - \epsilon_\alpha(b)\big)\left(\mathbb{E}_{j>b}[\log j] - \mathbb{E}_{j\le b}[\log j]\right) < 0, \tag{4}$$

since every tail index exceeds every retained index. Thus $\epsilon_\alpha(b)$ also decreases with α. □

Remark. Proposition 1 shows that, within the assumed power-law spectral family, flatter spectra with higher effective rank incur larger relative reconstruction error under the same rank budget. This provides theoretical motivation for using effective rank as a signal for adaptive budget allocation. It does not imply that the proportional rule in Eq. 2 is the exact reconstruction-error-minimizing allocation. We adopt Eq. 2 as a simple practical rule, whose effectiveness is validated in Section 5.4.

Efficient Approximation. Computing the exact effective rank requires obtaining the singular-value spectrum of each $\mathbf{V}_i$. Let $m_i = \min(n_i, d)$. Using the smaller Gram matrix requires $\mathcal{O}(n_i d m_i)$ computation for matrix multiplication followed by $\mathcal{O}(m_i^3)$ eigendecomposition. We instead use the *participation ratio* (PR) (Gao et al., 2017), which avoids eigendecomposition. Given the Gram matrix $\mathbf{G}_i = \mathbf{V}_i \mathbf{V}_i^\top$ (or $\mathbf{V}_i^\top \mathbf{V}_i$ when $n_i > d$), PR is defined as

$$\mathrm{PR}(\mathbf{V}_i) = \frac{\mathrm{tr}(\mathbf{G}_i)^2}{\|\mathbf{G}_i\|_F^2} = \frac{(\sum_j \lambda_j)^2}{\sum_j \lambda_j^2}, \tag{5}$$

where λ_j are the eigenvalues of $\mathbf{G}_i$. Like effective rank, PR is scale invariant and increases as the spectrum becomes more dispersed. We replace the entropy-based effective rank in Eq. 2 with PR as an efficient surrogate for budget allocation. While the two statistics are not guaranteed to induce identical rankings for arbitrary spectra, our ablation in Section 5.4 shows that PR and exact effective rank yield nearly identical performance. PR removes the $\mathcal{O}(m_i^3)$ eigendecomposition required to recover the full spectrum, leaving the Gram-matrix computation as the main cost.

4.2 Dynamism-Aware Token Selection (DATS)

Dynamism-Aware Score. With per-segment budgets determined, we next decide which tokens to retain within each segment. Existing methods primarily rely on `[CLS]` attention, which captures intra-frame importance but misses inter-frame temporal change. However, video is inherently temporal, and informative content lies in how visual features change across frames. We therefore introduce a dynamism score to complement attention-based saliency for measuring token importance.

For each token $v_j^{(t)}$ at frame t, let $v_{\text{nn}}^{(t\pm1)}$ denote its nearest neighbor in frame $t \pm 1$ by cosine similarity, and define $s^\pm = \mathrm{sim}(v_j^{(t)}, v_{\text{nn}}^{(t\pm1)})$. The dynamism score is:

$$\mathrm{dyn}(v_j) = \mathrm{novelty}(v_j) + \mathrm{asym}(v_j), \tag{6}$$

$$\mathrm{novelty}(v_j) = 1 - \max\big(s^-,\ s^+\big), \tag{7}$$

$$\mathrm{asym}(v_j) = \big|s^- - s^+\big|. \tag{8}$$

Novelty is high when a token undergoes substantial feature change across adjacent frames, while asymmetry captures differences between its forward and backward temporal correspondence. For boundary frames, novelty is computed using the available adjacent frame, while the asymmetry term is set to zero.

The final importance score combines attention-based saliency with the dynamism score:

$$q_j = \text{dyn}(v_j) + a_j, \tag{9}$$

where a_j denotes the normalized attention-based saliency score. Before combination, the attention and dynamism scores are independently normalized via z-score normalization.

Facility Location-Based Anchor Selection. With importance scores assigned, we select B_i anchor tokens per segment. A naive top-k selection based solely on individual importance may concentrate selected tokens in similar regions and fail to cover diverse segment content. We therefore formulate anchor selection as a Facility Location (FL) problem (Nemhauser et al., 1978) that jointly considers token importance and content coverage.

Given tokens with importance scores $\{q_k\}_{k=1}^{n_i}$, we define the weighted similarity between token k and candidate anchor j as

$$w_{kj} = q_k \cdot \cos(v_k, v_j), \tag{10}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity. We then seek a subset S of size B_i that maximizes

$$F(S) = \sum_{k=1}^{n_i} \max\left(0, \max_{j \in S} w_{kj}\right), \tag{11}$$

where n_i is the number of tokens in segment g_i. Intuitively, the importance score q_k determines how strongly token k contributes to the coverage objective, while the similarity term encourages each token to be represented by a semantically similar anchor. Maximizing $F(S)$ therefore favors anchors that jointly cover important and diverse segment content.

The objective $F(S)$ is monotone submodular (Nemhauser et al., 1978); under the cardinality constraint $|S| = B_i$, greedy selection achieves a $(1 - 1/e)$ approximation. At each step, we select

$$j^* = \arg\max_{j \notin S} \sum_{k=1}^{n_i} \max\big(w_{kj} - c_k,\ 0\big), \tag{12}$$

where $c_k = \max(0, \max_{j \in S} w_{kj})$ denotes the current coverage of token k, with $c_k = 0$ initially. After selecting j^*, the cached coverage is updated as $c_k \leftarrow \max(c_k, w_{kj^*})$.

Soft Fusion. The B_i selected anchors form the compressed segment representation, but non-anchor tokens may still carry useful information. To preserve it, we assign each non-anchor token v_j to its top-K most similar anchors with softmax weights:

$$w_{j \to c} = \frac{\exp(\cos(v_j, v_c))}{\sum_{c' \in \mathcal{N}_j} \exp(\cos(v_j, v_{c'}))}, \tag{13}$$

where $\mathcal{N}_j$ denotes the top-K anchors of token v_j. Each anchor is then fused with the weighted centroid of its assigned non-anchor tokens:

$$\hat{v}_c = \gamma v_c + (1 - \gamma) \frac{\sum_{j:\, c \in \mathcal{N}_j} w_{j \to c} v_j}{\sum_{j:\, c \in \mathcal{N}_j} w_{j \to c}}, \tag{14}$$

where γ balances the original anchor representation and the aggregated non-anchor context. This preserves non-anchor information without increasing the final token count.

Computational Complexity. FL-based anchor selection operates independently within each segment. The pairwise similarity computation, excluding greedy selection, costs $\mathcal{O}(\sum_{i=1}^{G} n_i^2 d)$ across G segments. Detailed algorithmic and complexity analyses are provided in Appendix B, while empirical overhead is reported in Section 5.3.

5 EXPERIMENTS

We conduct extensive experiments to evaluate DynVid from four perspectives: (1) compression performance across diverse video understanding benchmarks, VLLMs, and retention ratios (Section 5.2); (2) computational efficiency (Section 5.3); (3) the contributions of the main components through ablation studies (Section 5.4); and (4) performance with increased numbers of sampled frames and under fixed token budgets (Appendix D.2). We also report results on Qwen3-VL in Appendix D.1, hyperparameter sensitivity in Appendix E, and qualitative visualizations in Appendix F.

5.1 EXPERIMENTAL SETTING

Datasets We evaluate DynVid on eight video understanding benchmarks: Video-MME (Fu et al., 2025a), EgoSchema (Mangalam et al., 2023), LongVideoBench (Wu et al., 2024), MVBench (Li et al., 2024b), MLVU test and dev (Zhou et al., 2025), LVBench (Wang et al., 2025), and TempCompass (Liu et al., 2024). All use multiple-choice evaluation and collectively span short to hour-long videos, egocentric and third-person views, and diverse temporal reasoning tasks. Detailed descriptions are in Appendix C.1.

Baselines We compare DynVid with five representative training-free visual token compression methods: FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025), PruneVID (Huang et al., 2025), FastVID (Shen et al., 2025), and FlashVID (Fan et al., 2026). Detailed model configurations are provided in Appendix C.2.

Implementation Details We apply DynVid to four VLLMs: LLaVA-OneVision-7B (Li et al., 2024a), LLaVA-Video-7B (Zhang et al., 2024), Qwen2.5-VL-7B (Qwen Team, 2025a), and Qwen3-VL-8B (Qwen Team, 2025b). LLaVA-OneVision uniformly samples 32 frames ($32 \times 196 = 6{,}272$ tokens), and LLaVA-Video samples 64 frames ($64 \times 169 = 10{,}816$ tokens). Qwen2.5-VL and Qwen3-VL adopt dynamic resolution encoding with 32 sampled frames under default preprocessing. We evaluate four retention ratios, $\bar{R} \in \{20\%, 10\%, 7.5\%, 5\%\}$. All experiments are conducted in training-free manner using LMMs-Eval (Zhang et al., 2025a), without fine-tuning or additional learnable parameters. Following FlashVID (Fan et al., 2026), we derive attention-based token saliency from final-layer vision-encoder attention matrix. For encoders without an explicit `[CLS]` token, we compute per-token saliency by averaging the attention received by each visual token over all attention heads and query positions. We also follow FlashVID to align different methods by the average retention ratio $\bar{R}$ across Transformer layers. Additional details are provided in Appendix C.

5.2 MAIN RESULTS

Results on LLaVA-OneVision and LLaVA-Video. As shown in Table 1, DynVid achieves the best or competitive average performance across all four retention ratios on both models. On LLaVA-OneVision, DynVid attains 100.0% relative accuracy at $\bar{R}$=20%, matching the uncompressed vanilla model while using only one-fifth of the visual tokens, confirming that substantial token redundancy exists in video representations. At $\bar{R}$=10%, DynVid retains 99.3% relative accuracy, outperforming FlashVID by 1.3%. The advantage becomes more pronounced under aggressive compression: at $\bar{R}$=7.5% and 5%, DynVid achieves 98.2% and 96.4% relative accuracy, surpassing FlashVID by 1.8% in both cases. On LLaVA-Video (64 frames), DynVid ties for the best average at $\bar{R}$=20% (98.3% Rel. Acc, matching FastVID) and stays highly competitive at $\bar{R}$=10% (96.6% vs. FlashVID's 96.9%). At more aggressive ratios, DynVid clearly leads: it achieves 95.7% and 93.0% relative accuracy at $\bar{R}$=7.5% and 5%, outperforming FlashVID by 1.4% and 0.9% respectively. These results confirm that DynVid's advantage is more pronounced under tighter budgets, where adaptive allocation and dynamism-aware selection become increasingly critical.

Results on Qwen2.5-VL. To verify generalization beyond the LLaVA family, we evaluate on Qwen2.5-VL, which uses a substantially different tokenization pipeline (dynamic resolution with built-in spatial pooling). As shown in Table 1, DynVid consistently achieves the best results across all four retention ratios, with particularly large margins. At $\bar{R}$=10%, DynVid reaches 97.4% relative accuracy, surpassing FlashVID by 1.8%. At $\bar{R}$=5%, the gap widens to 1.9% (94.6% vs. 92.7%). These results confirm that DynVid is architecture-agnostic and not tied to a specific VLLM family.

Table 1: **Average performance on *LLaVA-OneVision*, *LLaVA-Video* and *Qwen2.5-VL*.** Avg Score (eight benchmarks) and Rel. Acc (%) under different retention ratios.

Method	$\bar{R} = 20\%$		$\bar{R} = 10\%$		$\bar{R} = 7.5\%$		$\bar{R} = 5\%$	
	Avg score	Rel. Acc(%)	Avg score	Rel. Acc(%)	Avg score	Rel. Acc(%)	Avg score	Rel. Acc(%)
LLaVA-OneVision, 32 frames (Vanilla: 56.0)								
VisionZip	54.9	98.0	51.8	92.5	48.7	87.0	46.4	82.9
PruneVID	53.4	95.4	52.1	93.0	51.1	91.3	49.8	88.9
FastVID	55.3	98.8	53.7	95.9	52.9	94.5	51.3	91.6
FlashVID	55.8	99.6	54.9	98.0	54.0	96.4	53.0	94.6
DynVid	**56.0**	**100.0**	**55.6**	**99.3**	**55.0**	**98.2**	**54.0**	**96.4**
LLaVA-Video, 64 frames (Vanilla: 58.2)								
VisionZip	55.2	94.8	52.6	90.4	50.3	86.4	44.8	77.0
PruneVID	54.9	94.3	52.7	90.5	51.5	88.5	49.7	85.4
FastVID	**57.2**	**98.3**	55.3	95.0	54.6	93.8	52.8	90.7
FlashVID	56.9	97.8	**56.4**	**96.9**	54.9	94.3	53.6	92.1
DynVid	**57.2**	**98.3**	56.2	96.6	**55.7**	**95.7**	**54.1**	**93.0**
Qwen2.5-VL, 32 frames (Vanilla: 57.5)								
VisionZip	56.0	97.4	54.4	94.6	53.2	92.5	51.4	89.3
FastVID	55.3	96.2	53.6	93.3	52.5	91.2	51.0	88.8
FlashVID	56.3	98.0	55.0	95.6	54.4	94.6	53.3	92.7
DynVid	**56.7**	**98.7**	**56.0**	**97.4**	**55.2**	**96.0**	**54.4**	**94.6**

Table 2: **Efficiency of our DynVid.** We conduct the efficiency analysis on **LLaVA-OneVision** and **Qwen2.5-VL**, and report the prefilling time and Time-To-First-Token (TTFT) in milliseconds (ms).

Method	**Retention Ratio** $\bar{R}$	**TFLOPs**	**Vision Encoding**	**Prefilling Time**			**TTFT**	**Avg.**	
				Compression	LLM Forward	Total		Score	Rel.(%)
LLaVA-OneVision									
Vanilla	100%	48.8	393.7	0.0	752.3	752.3	1146.0	56.0	100.0
FlashVID	10%	4.27	393.7	52.3	94.1	146.4 (5.14×)	540.1 (2.12×)	54.9	98.0
DynVid	10%	4.27	393.7	51.9	91.9	143.8 (5.23×)	537.5 (2.13×)	55.6	99.3
DynVid	7.5%	3.23	393.7	42.5	74.3	116.8 (6.44×)	510.5 (2.24×)	55.0	98.2
Qwen2.5-VL									
Vanilla	100%	101.8	755.4	0.0	1509.7	1509.7	2265.1	57.5	100.0
FlashVID	10%	7.82	755.4	49.5	152.1	201.6 (7.49×)	957.0 (2.37×)	55.0	95.6
DynVid	10%	7.82	755.4	53.1	147.3	200.4 (7.53×)	955.8 (2.37×)	56.0	97.4
DynVid	7.5%	5.81	755.4	43.9	115.1	159.0 (9.49×)	914.4 (2.48×)	55.2	96.0

5.3 Efficiency Analysis.

Token compression improves inference efficiency, but the compression itself introduces overhead. We evaluate prefilling time and TTFT on VideoMME, comparing DynVid with FlashVID (Table 2). On LLaVA-OneVision, DynVid at $\bar{R}$=10% achieves 99.3% Rel. Acc with 5.23× prefilling speedup, while FlashVID at the same ratio reaches only 98.0% with a comparable 5.14× speedup. At a more aggressive $\bar{R}$=7.5%, DynVid still attains 98.2% Rel. Acc, surpassing FlashVID at $\bar{R}$=10%, while achieving 6.44× prefilling and 2.24× TTFT speedup. On Qwen2.5-VL, DynVid at $\bar{R}$=7.5% surpasses FlashVID at $\bar{R}$=10% in both accuracy (96.0% vs. 95.6%) and speed (9.49× vs. 7.49× prefilling), delivering better quality with fewer tokens. At $\bar{R}$=10%, DynVid retains 97.4% Rel. Acc with 7.53× prefilling and 2.37× TTFT speedup over the Vanilla baseline.

5.4 Ablation analysis

Ablation on Adaptive Budget Allocation. Our adaptive allocation uses the participation ratio (PR) as an efficient surrogate for effective rank to quantify each segment's content complexity and distributes the token budget proportionally. To validate adaptive allocation, we replace it with a uniform baseline (w/ Uniform) that assigns a fixed per-frame budget, making allocation purely proportional to frame count. As shown in Table 3, adaptive allocation consistently outperforms the uniform variant, by 0.7% on LLaVA-OneVision (99.1% vs. 98.4%) and 1.1% on Qwen2.5-VL (97.6% vs. 96.5%). Uniform allocation over-spends on low-complexity segments while under-serving high-complexity segments that require more tokens to preserve visual information.

Table 3: Component ablation on four benchmarks.

Method	Video-MME	LongVideoBench	LVBench	TempCompass	Avg. score	Rel. Acc(%)
LLaVA-OneVision ($\bar{R} = 10\%$)						
Vanilla	58.5	56.6	38.7	64.8	54.65	100.0
w/ Uniform	57.7	55.9	37.4	64.1	53.78	98.4
w/ Exact eRank	58.1	56.7	37.6	64.2	54.15	99.1
w/o Dyn	57.7	55.9	38.0	63.9	53.88	98.5
w/o `[CLS]`	57.4	55.6	36.9	63.4	53.33	97.4
w/o FL	57.2	54.9	35.9	64.1	53.03	96.9
DynVid	57.7	56.5	38.2	64.3	**54.18**	**99.1**
Qwen2.5-VL ($\bar{R} = 10\%$)						
Vanilla	61.3	58.9	37.4	73.4	57.75	100.0
w/ Uniform	58.2	57.2	37.1	70.7	55.80	96.5
w/ Exact eRank	58.4	58.1	37.6	71.0	56.28	97.4
w/o Dyn	58.6	57.1	37.1	70.6	55.85	96.7
w/o `[CLS]`	58.1	57.8	37.2	70.4	55.88	96.7
w/o FL	58.0	56.4	36.5	70.4	55.33	95.7
DynVid	58.4	58.2	37.8	71.1	**56.38**	**97.6**

We further replace PR with the exact entropy-based effective rank in Eq. 1 (w/ Exact eRank). The two variants achieve nearly identical performance: 54.18 vs. 54.15 average score on LLaVA-OneVision and 56.38 vs. 56.28 on Qwen2.5-VL. This indicates that, although PR is not guaranteed to induce the same ordering as exact effective rank for arbitrary spectra, it serves as an effective practical surrogate in our setting.

Ablation on Dynamism Integration. Our importance scoring (Eq. 9) combines attention score with dynamism signal that captures inter-frame temporal change. We evaluate two variants: (i) w/o Dyn, which sets $\text{dyn}(v_j) = 0$ and relies solely on `[CLS]` attention; and (ii) w/o `[CLS]`, which removes attention score and uses only dynamism. As shown in Table 3, the full combination outperforms both single-signal variants. Removing dynamism reduces relative accuracy by 0.6% on LLaVA-OneVision (99.1% vs. 98.5%) and 0.9% on Qwen2.5-VL (97.6% vs. 96.7%); removing `[CLS]` attention leads to larger drops of 1.7% and 0.9% respectively. This confirms that the two signals are complementary: `[CLS]` attention captures spatial saliency, while the dynamism signal emphasizes tokens undergoing significant inter-frame feature change.

Ablation on FL-Based Anchor Selection. Our anchor selection employs a Facility Location (FL) objective to jointly optimize importance and content coverage within each segment. We replace FL with a top-k baseline (w/o FL) that greedily picks the B_i highest-importance tokens without any coverage constraint, keeping all other components unchanged. As shown in Table 3, FL-based selection outperforms top-k by 2.2% on LLaVA-OneVision (99.1% vs. 96.9%) and 1.9% on Qwen2.5-VL (97.6% vs. 95.7%). This confirms that importance alone is insufficient: top-k concentrates anchors in a few salient regions, losing informative spatial-temporal context elsewhere. The FL objective explicitly penalizes such redundant selections, spreading anchors to cover the full segment content.

6 Conclusion

In this work, we propose DynVid, a training-free, dynamism-aware adaptive token compression framework for Video LLMs. DynVid follows an *allocate-then-select* paradigm: Effective Rank-based Adaptive Allocation (ERA) estimates segment complexity via participation ratio to dynamically distribute token budgets, while Dynamism-Aware Token Selection (DATS) combines inter-frame dynamism with `[CLS]` saliency and Facility Location optimization to select diverse, representative anchor tokens. Experiments on eight benchmarks and four VLLMs show strong performance under 5%–20% token retention with significant prefilling speedup.

Reproducibility Statement

To ensure reproducibility, we provide detailed implementation settings and experimental protocols in Section 5.1 and Appendix C. We also release the code at the anonymous repository: `https://anonymous.4open.science/r/DynVid`.

AI Use Statement

In this work, we used generative AI tools to assist with manuscript editing and technical review, including improving language, clarity, organization, and methodological and experimental descriptions, as well as refining the presentation and verification of mathematical arguments and proofs. Generative AI tools were not used to generate experimental results. All AI-assisted content, including mathematical derivations and technical descriptions, was carefully reviewed, verified, and revised by the authors. We take full responsibility for the final content of this work.

Ethics Statement

This work focuses on methodological research for efficient Video LLM inference and does not involve human subjects or the collection of personally identifiable information. Experiments are conducted on established benchmark datasets and pretrained models without introducing new data collection. The proposed token compression method may be applied to real-world video understanding tasks, where information loss or prediction errors could affect downstream decisions; therefore, deployment in high-stakes settings should include appropriate domain-specific validation and safeguards.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9392–9401, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations (ICLR)*, 2023.

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 *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024c.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36: 49250–49267, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

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 *International Conference on Learning Representations (ICLR)*, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22654–22663, 2025b.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *BioRxiv*, pp. 214262, 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

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. *arXiv preprint arXiv:2408.03326*, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024b.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102, 2025.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:34892–34916, 2023.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024.

Zhihang Liu, Chen-Wei Xie, Pandeng Li, Liming Zhao, Longxiang Tang, Yun Zheng, Chuanbin Liu, and Hongtao Xie. Hybrid-level instruction injection for video token compression in multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8568–8578, 2025.

Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:46212–46244, 2023.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Qwen Team. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025a.

Qwen Team. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, and Jie Tang. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning (ICML)*, 2025b.

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

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. Aim: Adaptive inference of multi-modal llms via token merging and pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20180–20192, 2025.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

A Extended Related Work

We provide a more comprehensive discussion of related work below.

A.1 Multimodal Large Language Models

Recent advances in large language models (LLMs) (Touvron et al., 2023; OpenAI, 2023) have driven the development of multimodal large language models (MLLMs) that integrate visual perception with language reasoning. Early image-oriented MLLMs such as LLaVA (Liu et al., 2023), InstructBLIP (Dai et al., 2023), and mPLUG-Owl (Ye et al., 2023) demonstrated strong image understanding by aligning a frozen visual encoder with an LLM through lightweight projection modules. Extending MLLMs from images to videos, recent works encode sampled frames into visual tokens and feed them to the LLM backbone. Representative models include LLaVA-OneVision (Li et al., 2024a), LLaVA-Video (Zhang et al., 2024), InternVL 1.5 (Chen et al., 2024b), and Video-LLaVA (Lin et al., 2024), which encode each frame independently and concatenate the resulting tokens. Others such as Qwen2-VL (Wang et al., 2024) and Qwen2.5-VL (Qwen Team, 2025a) integrate temporal modeling directly into the visual encoder through 3D convolutions. Video-LLaVA (Lin et al., 2024) and VideoChat2 (Li et al., 2024b) further explore unified visual representations that bridge image and video modalities. Despite architectural differences, these approaches share a common challenge: the number of visual tokens grows linearly with the number of input frames, creating substantial computational cost during LLM inference and an urgent need for efficient token compression strategies.

A.2 Visual Token Compression for Images

Before addressing the video setting, we first review token compression techniques developed for image-based MLLMs, as they form the foundation for many video methods. **Attention-based pruning** methods leverage the LLM's own attention mechanism to identify dispensable tokens. FastV (Chen et al., 2024a) computes attention scores at a designated layer and removes the bottom-K tokens, observing that many image tokens receive negligible attention after the first few decoder layers. SparseVLM (Zhang et al., 2025b) selects visual-relevant text tokens to rate the importance of visual tokens within the self-attention matrix, and progressively prunes less important ones with a rank-based sparsification strategy. It further introduces a token recycling method that compresses pruned tokens into compact representations. PyramidDrop (Xing et al., 2025) applies increasingly aggressive pruning at deeper layers, motivated by the observation that visual token redundancy progressively increases in deeper layers. DivPrune (Alvar et al., 2025) formulates token pruning as a Max-Min Diversity Problem (MMDP), ensuring that retained tokens cover a broad spatial extent rather than clustering around a few salient regions. **Merging-based methods** avoid hard information loss by combining redundant tokens. Token Merging (ToMe) (Bolya et al., 2023) partitions tokens into source and destination sets via bipartite matching on cosine similarity, then merges source tokens into their matched destinations. VisionZip (Yang et al., 2025) selects dominant tokens based on attention scores from the visual encoder, and merges the remaining tokens into contextual tokens based on semantic similarity. AIM (Zhong et al., 2025) unifies merging and pruning into a single training-free adaptive inference framework that accommodates different efficiency requirements.

While effective for images, these methods treat each frame independently when applied to video, failing to exploit temporal redundancy or distribute budgets according to content complexity.

A.3 Visual Token Compression for Video

To address these limitations, several methods have been proposed to explicitly exploit temporal redundancy across frames. PruneVid (Huang et al., 2025) first merges spatiotemporal tokens to reduce inter-frame redundancy, then leverages question-guided attention from the LLM to selectively prune visual tokens irrelevant to the query. DyCoke (Tao et al., 2025) operates at two complementary stages: a plug-and-play temporal merging module reduces redundant tokens across consecutive frames before the LLM, while dynamic KV cache pruning inside the LLM further removes spatially redundant tokens during decoding. FastVID (Shen et al., 2025) dynamically partitions videos into temporally ordered segments and applies density-based token pruning within each segment, though it distributes budgets uniformly without considering per-segment complexity. FrameFu-

sion (Fu et al., 2025b) merges tokens with high inter-frame similarity at shallow layers, followed by importance-based pruning at deeper layers in a cascaded manner. FlashVID (Fan et al., 2026) introduces a tree-based spatiotemporal token merging mechanism that jointly exploits spatial and temporal redundancy before LLM ingestion, complemented by attention and diversity-based selection (ADTS) that ensures feature coverage within each frame. HoliTom (Shao et al., 2025) integrates global redundancy-aware temporal segmentation with outer-LLM spatiotemporal merging and inner-LLM similarity-based merging. HICom (Liu et al., 2025) takes a different angle by injecting text instructions as a condition for compression at both local and global levels, selectively preserving tokens relevant to the query.

Despite these advances, existing video methods still exhibit two key limitations: (1) they apply uniform or heuristic budgets across segments without adapting to content complexity, and (2) their importance signals, whether attention-based or diversity-based, operate within individual frames and do not account for the temporal dynamics of video content. DynVid addresses both gaps through adaptive budget allocation based on effective rank estimation and dynamism-aware importance scoring that captures how visual content evolves over time.

B Additional Details of Anchor Selection

We provide the complete pseudocode for our Facility Location-based anchor selection in Algorithm 1. Stage 1 greedily selects B_i anchor tokens by maximizing weighted content coverage, while Stage 2 fuses non-anchor information into the selected anchors via softmax-weighted aggregation.

Algorithm 1 FL-Based Anchor Selection and Soft Fusion

Require: Tokens $\{v_j\}_{j=1}^{n_i}$, scores $\{q_j\}$, budget B_i, fusion weight γ, top-K
Ensure: B_i compressed tokens

// FL Anchor Selection
$w_{kj} \leftarrow q_k \cos(v_k, v_j)$ for all k, j; $c_k \leftarrow 0$; $S \leftarrow \emptyset$
for $b = 1, \ldots, B_i$ **do**
  $\Delta_j \leftarrow \sum_{k=1}^{n_i} \max(w_{kj} - c_k, 0)$ for all $j \notin S$
  $j^* \leftarrow \arg\max_{j \notin S} \Delta_j$; $S \leftarrow S \cup \{j^*\}$; $c_k \leftarrow \max(c_k, w_{kj^*})$
end for
$\mathcal{A} \leftarrow S$, $\mathcal{R} \leftarrow \{1, \ldots, n_i\} \setminus \mathcal{A}$
// Soft Fusion
$m_c \leftarrow 0$, $z_c \leftarrow 0$ for all $c \in \mathcal{A}$
for each $j \in \mathcal{R}$ **do**
  Find top-K anchors $\mathcal{N}_j \subseteq \mathcal{A}$ by $\cos(v_j, v_c)$
  $w_{j \to c} \leftarrow \frac{\exp(\cos(v_j, v_c))}{\sum_{c' \in \mathcal{N}_j} \exp(\cos(v_j, v_{c'}))}$, $c \in \mathcal{N}_j$
  $m_c \mathrel{+}= w_{j \to c} v_j$, $z_c \mathrel{+}= w_{j \to c}$ for $c \in \mathcal{N}_j$
end for
$\hat{v}_c \leftarrow \gamma v_c + (1 - \gamma) m_c / z_c$ if $z_c > 0$, otherwise $\hat{v}_c \leftarrow v_c$
return $\{\hat{v}_c \mid c \in \mathcal{A}\}$

Computational Complexity. FL-based anchor selection is performed independently within each segment. For segment g_i with n_i tokens and feature dimension d, constructing the weighted similarity matrix costs $\mathcal{O}(n_i^2 d)$ computation and $\mathcal{O}(n_i^2)$ memory. Greedy selection performs B_i iterations, each evaluating marginal gains for all candidates over n_i tokens, resulting in $\mathcal{O}(B_i n_i^2)$ computation. Thus, the overall FL selection cost is $\mathcal{O}(n_i^2(d + B_i))$ per segment and

$$\mathcal{O}\left(\sum_{i=1}^{G} n_i^2 (d + B_i)\right) \tag{15}$$

across G segments, with $\mathcal{O}(n_i^2)$ memory per segment. Temporal segmentation confines the quadratic computation to individual segments rather than all video tokens jointly. Cached coverage and GPU-parallel marginal evaluation reduce practical overhead without changing the asymptotic complexity.

C Implementation Details

C.1 Evaluation Benchmarks

We evaluate DynVid on eight diverse video understanding benchmarks that collectively span a wide range of video lengths, domains, and reasoning complexities.

Video-MME (Fu et al., 2025a) is a comprehensive multi-modal evaluation benchmark covering 6 major visual domains and 30 subfields, with a total of 900 videos and 2,700 multiple-choice questions. It features videos of varying durations categorized into short (11s–2min), medium (4min–15min), and long (30min–60min) subsets, enabling systematic analysis of model performance across different temporal scales.

EgoSchema (Mangalam et al., 2023) is a diagnostic benchmark derived from Ego4D (Grauman et al., 2022), consisting of 5,031 human-curated multiple-choice questions based on three-minute egocentric video clips. Each question provides five answer options and requires holistic understanding of natural human activities, including temporal ordering, causal reasoning, and activity summarization over extended temporal contexts. We evaluate on the full set.

LongVideoBench (Wu et al., 2024) contains 3,763 web-collected videos with subtitles and 6,678 human-annotated multiple-choice questions spanning 17 categories, with video-language interleaved inputs up to an hour long. Its distinctive referring reasoning task requires models to retrieve and reason over detailed multimodal information from specific temporal segments referenced by natural language queries, testing both long-range temporal understanding and precise content localization. We evaluate on the validation set in our experiments.

MVBench (Li et al., 2024b) provides 20 challenging temporal video understanding tasks that cannot be effectively solved with a single frame, covering action recognition, object tracking, causal reasoning, and other temporal reasoning abilities. The benchmark contains 4,000 multiple-choice questions with videos primarily ranging from 5 to 60 seconds. All tasks are specifically designed to require temporal reasoning across multiple frames, making it particularly suitable for assessing fine-grained temporal comprehension.

MLVU (Zhou et al., 2025) is a multi-task long video understanding benchmark designed to evaluate MLLMs on videos with substantial lengths. It covers diverse video genres and includes multiple evaluation tasks across multiple-choice and generation formats. The benchmark features videos ranging from 3 minutes to over 2 hours, testing abilities such as topic reasoning, anomaly recognition, and plot summarization. We evaluate on both the development set (MLVU-dev) and the test set (MLVU-test).

LVBench (Wang et al., 2025) is a benchmark specifically designed for extreme long video understanding. It comprises 103 publicly sourced videos (each exceeding 30 minutes, averaging approximately 68 minutes) with 1,549 question-answer pairs that test six core capabilities: temporal grounding, summarization, reasoning, entity recognition, event understanding, and key information retrieval. The benchmark challenges models to demonstrate long-term memory and extended comprehension capabilities over significantly longer temporal contexts than typical video benchmarks.

TempCompass (Liu et al., 2024) is a benchmark for evaluating the temporal perception ability of Video LLMs. It introduces a diversity of temporal aspects (e.g., speed, direction, event order) and task formats. The benchmark constructs conflicting videos that share the same static content but differ in specific temporal aspects, preventing models from leveraging single-frame bias or language priors. We evaluate on the multiple-choice QA task.

All benchmarks use accuracy on multiple-choice questions as the evaluation metric, implemented through LMMs-Eval (Zhang et al., 2025a).

C.2 Baseline Methods

We compare DynVid with five representative training-free visual token compression methods across four VLLMs (LLaVA-OneVision (Li et al., 2024a), LLaVA-Video (Zhang et al., 2024), Qwen2.5-VL (Qwen Team, 2025a), and Qwen3-VL (Qwen Team, 2025b)). Since our method is fully deterministic at inference time, all reported results are from a single run. All baselines are re-implemented

in LMMs-Eval (Zhang et al., 2025a) following their official hyperparameter configurations. Results are directly taken from FlashVID (Fan et al., 2026) where available; for additional settings not covered therein, we reproduce results using the same configurations. Below we describe each baseline and its hyperparameter settings for reproducibility.

FastV (Chen et al., 2024a) (ECCV 2024) prunes tokens at the K-th layer of the LLM using cross-modal attention scores, with a pruning ratio r. We set $K = 2$, using $r \in \{80\%, 90\%, 92.5\%, 95\%\}$ for all VLLMs.

VisionZip (Yang et al., 2025) (CVPR 2025) selects dominant tokens and merges contextual tokens at the output of the vision encoder. We apply pruning after pooling, retaining dominant and contextual tokens at a 54:10 ratio per frame. We set $\bar{R}$ to $\{20\%, 10\%, 7.5\%, 5\%\}$ for all VLLMs.

PruneVID (Huang et al., 2025) (ACL 2025) combines before-LLM compression and inner-LLM pruning during prefilling, along with KV Cache compression at decoding. We set $\tau = 0.8$, $\gamma = 0.25$, $\alpha = 0.4$, $K = 10$, and adjust the cluster ratio β to match the target retention ratios $\{20\%, 10\%, 7.5\%, 5\%\}$.

FastVID (Shen et al., 2025) (NeurIPS 2025) performs dynamic segmentation followed by salient token selection and spatiotemporal redundancy elimination based on DPC-kNN. We set segments $c = 8$, threshold $\tau = 0.9$, salient token ratio $d = 0.4$, anchor frame step $p = 4$, and merging factor $\alpha = 0.6$. We set $\bar{R}$ to $\{20\%, 10\%, 7.5\%, 5\%\}$ for all VLLMs.

FlashVID (Fan et al., 2026) (ICLR 2026) applies attention and diversity-based token selection (ADTS) to select informative tokens per frame by solving a calibrated Max-Min Diversity Problem, then applies tree-based spatiotemporal token merging (TSTM) on the remaining tokens, which links each token to its most similar counterpart in the adjacent frame to build redundancy trees and aggregates them via mean pooling. We set the calibration weight $\alpha = 0.7$, merging threshold $T_\tau = 0.8$, and expansion factor $f_e = 1.25$. We set $\bar{R}$ to $\{20\%, 10\%, 7.5\%, 5\%\}$ for all VLLMs.

C.3 Token Budget Alignment

To ensure a fair comparison, we adopt the same two-stage token-budget protocol as FlashVID (Fan et al., 2026), consisting of before-LLM compression followed by inner-LLM query-guided pruning. Since our contribution focuses on the before-LLM compression stage, we follow FlashVID for the inner-LLM pruning configuration to isolate the effect of the before-LLM compression strategy.

Specifically, the token budget B denotes the number of visual tokens retained after before-LLM compression and entering the LLM. At layer K_L, query-guided pruning further reduces the visual token count from B to R for the remaining layers. The average retention ratio $\bar{R}$ is defined as the average fraction of visual tokens retained across Transformer layers:

$$\bar{R} = \frac{B \cdot K_L + R \cdot (L - K_L)}{L}, \tag{16}$$

where L is the total number of Transformer layers in the LLM.

Following FlashVID, we introduce an expansion factor f_e such that $B = f_e \cdot \bar{R} \cdot N$, which controls the number of tokens retained before the LLM as candidates for subsequent query-guided pruning. We set $f_e = 1.25$ in all experiments. For example, to achieve $\bar{R} = 10\%$ on Qwen2.5-VL 7B ($L = 28$), we have $B = f_e \cdot \bar{R} \cdot N = 1.25 \times 10\% \cdot N = 12.5\% \cdot N$ entering the LLM, with $R = 3.75\% \cdot N$ after inner-LLM pruning at layer $K_L = 20$.

The average retention ratio $\bar{R}$ provides a unified measure of layer-wise token usage across different compression methods. We follow FlashVID by fixing $f_e = 1.25$ and $K_L = 20$ throughout all experiments without further tuning.

D More Results

D.1 Full Per-Benchmark Results

We provide the full per-benchmark results that complement the average scores reported in the main paper.

Detailed Results on LLaVA-OneVision and LLaVA-Video. Table 4 presents the per-benchmark performance of all methods on LLaVA-OneVision and LLaVA-Video, respectively. DynVid achieves the best or highly competitive performance on the majority of individual benchmarks across all retention ratios.

Detailed Results on Qwen2.5-VL. Table 5 reports the per-benchmark results on Qwen2.5-VL. Despite the architectural differences (2D-RoPE and dynamic resolution encoding), DynVid maintains consistent improvements over all baselines, demonstrating its generalizability across different visual encoding paradigms.

Table 4: **Comparison of state-of-the-art methods on LLaVA-OneVision and LLaVA-Video.** Our DynVid outperforms previous state-of-the-art methods under different retention ratios $\bar{R}$ (20%, 10%, 7.5%, 5%) across multiple benchmarks. **Bold** denotes the best performance under each retention ratio. Avg. Score is the arithmetic mean across all benchmarks; Rel. Acc indicates the relative accuracy compared to the uncompressed vanilla model.

Method	Retention Ratio $\bar{R}$	VideoMME	EgoSchema	LongVideo Bench	MVBench	MLVU Test	MLVU Dev	LVBench	Temp Compass	Avg. Score	Avg. Rel. (%)
LLaVA-OneVision											
Vanilla	100%	58.5	60.3	56.6	58.3	47.7	63.1	38.7	64.8	56.0	100.0
VisionZip	20%	57.9	59.8	55.2	57.6	43.6	62.3	37.8	64.6	54.9	98.0
PruneVID		56.9	59.8	55.4	55.1	42.3	60.5	35.5	61.7	53.4	95.4
FastVID		57.4	59.2	56.9	**58.2**	44.7	61.3	**38.6**	**65.8**	55.3	98.8
FlashVID		57.9	60.1	**57.8**	58.2	46.2	63.6	36.9	65.8	55.8	99.6
DynVid		**58.1**	**60.4**	57.0	57.6	**47.4**	**64.3**	38.4	64.6	**56.0**	**100.0**
VisionZip	10%	52.9	58.3	50.2	54.5	41.3	59.0	36.2	61.6	51.8	92.5
PruneVID		54.7	59.1	52.3	53.3	42.6	57.8	36.3	60.8	52.1	93.1
FastVID		57.0	58.5	55.3	**57.2**	42.6	60.3	37.3	61.3	53.7	95.9
FlashVID		57.5	59.9	56.1	57.0	44.4	**62.7**	36.5	**64.8**	54.9	98.0
DynVid		**57.7**	**61.0**	**56.5**	57.1	**47.4**	62.2	**38.2**	64.3	**55.6**	**99.3**
VisionZip	7.5%	50.5	54.9	48.5	48.0	39.2	55.9	34.1	58.4	48.7	87.0
PruneVID		53.5	58.1	51.2	52.1	42.1	56.5	35.4	59.6	51.1	91.3
FastVID		55.0	58.3	53.8	56.4	43.7	59.1	35.7	61.3	52.9	94.5
FlashVID		56.2	60.0	54.5	56.2	43.9	**61.4**	36.6	63.5	54.0	96.4
DynVid		**56.6**	**60.8**	**55.5**	**57.4**	**46.7**	61.1	**38.1**	**63.7**	**55.0**	**98.2**
VisionZip	5%	48.3	51.5	46.7	45.4	36.5	54.3	32.8	55.8	46.4	82.9
PruneVID		52.5	57.8	50.1	51.3	39.6	55.7	34.9	56.8	49.8	88.9
FastVID		54.2	57.0	51.3	53.5	43.2	57.3	34.8	59.3	51.3	91.6
FlashVID		54.3	59.2	53.3	54.6	43.5	60.3	**36.5**	62.1	53.0	94.6
DynVid		**55.2**	**60.3**	**54.7**	**56.8**	**45.8**	**60.5**	36.1	**62.6**	**54.0**	**96.4**
LLaVA-Video											
Vanilla	100%	64.2	57.3	59.5	61.9	52.0	65.9	40.7	64.4	58.2	100.0
VisionZip	20%	61.3	**56.3**	56.6	56.0	49.0	63.6	38.6	60.1	55.2	94.8
PruneVID		59.1	54.8	57.4	56.6	47.4	63.2	37.9	62.5	54.9	94.3
FastVID		**63.1**	54.9	57.6	**60.7**	**50.0**	64.2	**41.1**	**66.3**	**57.2**	**98.3**
FlashVID		61.7	**56.3**	58.4	59.9	48.4	**64.8**	40.7	64.6	56.9	97.8
DynVid		61.6	**56.3**	**59.6**	60.2	**50.0**	64.4	39.9	65.4	**57.2**	**98.3**
VisionZip	10%	58.7	53.9	52.8	54.9	45.6	59.8	37.2	57.6	52.6	90.4
PruneVID		57.5	52.7	56.1	55.8	44.2	60.1	35.5	60.0	52.7	90.5
FastVID		60.6	53.6	56.8	59.6	46.5	61.2	39.7	64.5	55.3	95.0
FlashVID		60.9	**55.0**	57.5	59.7	**48.1**	**63.2**	**40.9**	66.0	**56.4**	**96.9**
DynVid		**61.2**	54.6	**58.1**	**59.8**	47.8	62.4	39.4	**66.3**	56.2	96.6
VisionZip	7.5%	56.7	51.8	50.2	52.4	40.5	59.1	35.6	56.1	50.3	86.4
PruneVID		56.7	51.9	53.5	53.6	42.0	59.1	35.7	59.7	51.5	88.5
FastVID		59.8	53.2	56.0	**59.3**	45.3	60.1	**39.6**	63.8	54.6	93.8
FlashVID		59.1	53.5	56.6	59.0	45.5	**62.5**	38.2	65.1	54.9	94.3
DynVid		**60.6**	**53.6**	**57.6**	**59.3**	**48.0**	61.2	**39.6**	**65.9**	**55.7**	**95.7**
VisionZip	5%	50.7	42.6	47.9	44.6	35.0	54.1	32.9	50.8	44.8	77.0
PruneVID		54.8	50.0	53.9	52.7	37.6	56.2	35.1	57.2	49.7	85.4
FastVID		58.2	51.8	53.4	57.8	43.7	57.8	37.6	62.4	52.8	90.7
FlashVID		58.2	52.1	54.8	57.8	43.3	**61.0**	**37.7**	64.0	53.6	92.1
DynVid		**59.9**	**52.9**	**55.6**	**58.2**	**45.7**	58.4	36.8	**65.5**	**54.1**	**93.0**

Results on Qwen3-VL. To further validate generality on the latest model architecture, we evaluate on Qwen3-VL (8B). As shown in Figure 3, DynVid achieves the best average performance across all four retention ratios, and the advantage over baselines grows as compression becomes more aggressive. At $\bar{R}$=10%, DynVid achieves 93.6% relative accuracy, surpassing FlashVID by 1.0% (93.6% vs. 92.6%). At $\bar{R}$=5%, the gap widens significantly to 2.3% (90.1% vs. 87.9%), demonstrating that well-calibrated allocation and precise token selection become increasingly critical as the token budget shrinks.

Table 5: **Comparison of state-of-the-art methods on Qwen2.5-VL.** Our DynVid outperforms previous state-of-the-art methods under different retention ratios $\bar{R}$ (20%, 10%, 7.5%, 5%) across multiple benchmarks.

Method	Retention Ratio $\bar{R}$	VideoMME	EgoSchema	LongVideo Bench	MVBench	MLVU Test	MLVU Dev	LVBench	Temp Compass	Avg. Score	Avg. Rel. (%)
Vanilla	100%	61.3	58.3	58.9	68.0	43.6	59.6	37.4	73.4	57.5	100.0
VisionZip	20%	58.8	56.6	57.1	66.5	41.3	59.2	37.0	71.5	56.0	97.4
FastVID		58.1	56.6	**58.5**	65.4	41.0	57.9	35.5	69.6	55.3	96.2
FlashVID		60.0	56.9	57.6	**67.4**	41.0	59.1	36.8	71.8	56.3	98.0
DynVid		**60.2**	**57.1**	58.0	66.9	**42.6**	**59.5**	**37.2**	**72.4**	**56.7**	**98.7**
VisionZip	10%	56.5	55.7	55.1	64.9	41.8	56.1	35.7	69.2	54.4	94.6
FastVID		55.9	55.4	56.0	63.5	39.6	56.7	35.7	66.2	53.6	93.3
FlashVID		57.7	55.9	56.5	65.1	41.5	57.3	36.4	69.5	55.0	95.6
DynVid		**58.4**	**56.1**	**58.2**	**65.5**	**42.5**	**58.6**	**37.8**	**71.1**	**56.0**	**97.4**
VisionZip	7.5%	55.2	54.8	53.5	64.1	40.1	55.5	34.5	67.6	53.2	92.5
FastVID		55.2	54.9	54.8	61.8	37.7	55.7	34.2	65.4	52.5	91.2
FlashVID		56.9	55.1	56.0	**65.0**	42.0	56.2	35.1	68.9	54.4	94.6
DynVid		**57.1**	**55.5**	**56.9**	**65.0**	**42.5**	**58.3**	**36.7**	**69.8**	**55.2**	**96.0**
VisionZip	5%	54.2	53.7	52.3	61.4	36.3	54.1	33.5	65.4	51.4	89.3
FastVID		53.9	53.3	54.0	59.3	36.8	54.4	34.0	62.7	51.0	88.8
FlashVID		56.0	**54.3**	55.0	**63.8**	39.8	55.7	34.5	67.2	53.3	92.7
DynVid		**56.4**	54.2	**56.8**	**63.8**	**42.4**	**57.6**	**35.2**	**68.6**	**54.4**	**94.6**

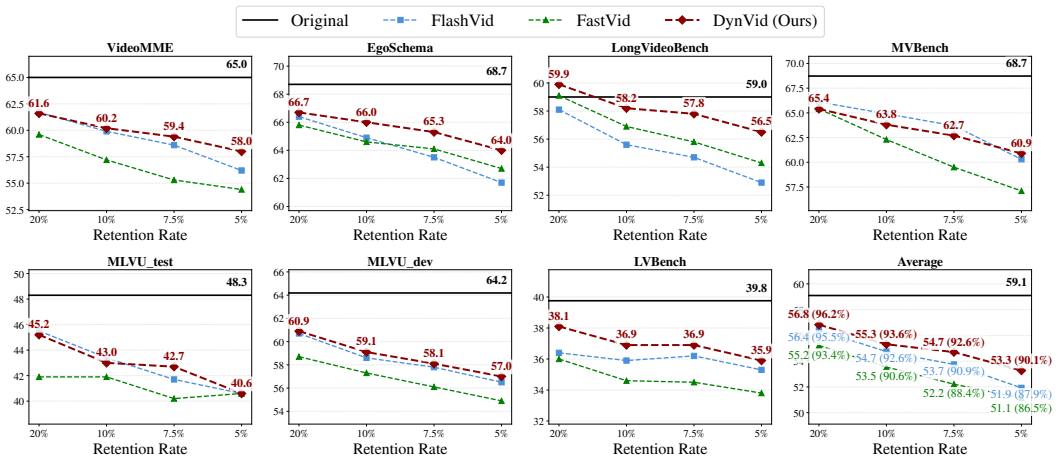


Figure 3: Comparison of SOTA methods on Qwen3-VL across eight benchmarks under retention ratios from 5% to 20%. DynVid outperforms FlashVID and FastVID at all compression levels.

D.2 Extended Analysis of Frame and Token Budgets

Results under Fixed Token Budget. Existing VLLMs typically sample only a limited number of frames for efficiency, potentially missing fine-grained details and important visual cues distributed across longer temporal spans. Token compression provides a natural remedy: by reducing total token count, it allows models to ingest more frames within the same computational budget, thereby capturing richer temporal context without increasing LLM prefilling time. To verify this, we fix the token budget to that of the vanilla 16-frame setting and scale the number of input frames to 32, 64, and 160, corresponding to token retention ratios of 50%, 25%, and 10%, respectively, allowing us to directly evaluate whether broader temporal coverage compensates for reduced per-frame detail, and to assess the model's ability to preserve fine-grained information and temporal cues under the same token budget.

As shown in Table 6, both LLaVA-OneVision and Qwen2.5-VL equipped with token compression methods generally maintain or improve performance over their vanilla 16-frame baselines, confirming that longer temporal context provides substantial gains even after aggressive per-frame compression. Among the compression methods, DynVid delivers the best Avg Score across all frame configurations on both models, outperforming FlashVID by 0.1%–1.5% on LLaVA-OneVision and 0.8%–1.8% on Qwen2.5-VL. Notably, the advantage of DynVid grows as the number of frames in-

Table 6: **Results under fixed token budget.** We fix the total visual token count to that of the vanilla 16-frame setting and increase the number of input frames via token compression. Token retention ratios are 50%, 25%, and 10% for 32, 64, and 160 frames, respectively. **Bold** denotes the best performance under each frame configuration.

Method	Frames (Retention $\bar{R}$)	VideoMME	EgoSchema	LongVideo Bench	MVBench	MLVU Test	MLVU Dev	LVBench	Temp Compass	Avg. Score	Avg. Rel. Acc (%)
LLaVA-OneVision											
Vanilla	16 (100%)	56.8	59.6	55.4	58.1	41.8	61.1	36.7	64.9	54.3	100.0
FastVID	32 (50%)	58.4	59.6	56.2	58.1	45.8	62.7	**39.7**	64.8	55.7	102.5
FlashVID		58.8	60.3	55.8	**58.2**	44.9	**64.0**	38.0	**65.6**	55.7	102.6
DynVid		**58.9**	**60.5**	**57.1**	57.6	**46.3**	63.4	37.1	65.4	**55.8**	**102.7**
FastVID	64 (25%)	58.3	59.5	57.2	57.0	46.5	64.8	37.9	64.2	55.7	102.5
FlashVID		58.9	60.0	**57.7**	**58.2**	47.5	65.0	37.8	**65.6**	56.3	103.8
DynVid		**59.1**	**60.7**	57.0	58.1	**48.9**	**66.2**	**39.0**	65.2	**56.8**	**104.6**
FastVID	160 (10%)	57.3	58.5	55.1	55.1	43.7	63.5	39.6	61.3	54.3	99.9
FlashVID		59.7	60.6	56.9	56.8	48.5	65.2	39.7	64.1	56.4	103.9
DynVid		**60.4**	**60.7**	**57.1**	**58.0**	**48.7**	**66.9**	**40.8**	**65.2**	**57.2**	**105.4**
Qwen2.5-VL											
Vanilla	16 (100%)	58.2	55.5	56.4	66.5	38.3	56.5	34.6	72.0	54.8	100.0
FastVID	32 (50%)	60.0	57.0	58.2	66.0	**43.5**	58.8	37.7	71.8	56.6	103.4
FlashVID		**61.7**	57.6	58.4	67.6	42.0	58.7	37.1	73.1	57.0	104.2
DynVid		60.9	**57.9**	**58.7**	**68.1**	43.1	**60.3**	**38.0**	**73.5**	**57.6**	**105.1**
FastVID	64 (25%)	61.7	57.9	58.2	65.4	45.3	61.1	38.1	70.6	57.3	104.6
FlashVID		62.0	58.3	**59.2**	**68.5**	45.5	62.2	38.2	73.0	58.4	106.6
DynVid		**62.2**	**58.4**	**59.2**	68.3	**46.9**	**62.6**	**38.8**	**74.0**	**58.8**	**107.4**
FastVID	160 (10%)	62.9	58.7	58.0	62.9	47.5	63.4	40.5	66.5	57.6	105.1
FlashVID		63.9	**59.5**	61.1	66.8	49.7	64.7	42.2	**73.4**	60.2	109.9
DynVid		**65.5**	59.4	**61.6**	**68.2**	**52.3**	**66.3**	**43.3**	72.8	**61.2**	**111.7**

creases (*e.g.*, from 0.1% at 32 frames to 1.5% at 160 frames on LLaVA-OneVision, and from 0.9% to 1.8% on Qwen2.5-VL), suggesting that DynVid is more effective at preserving informative tokens under higher compression ratios, where precise token selection becomes increasingly critical.

Extended Results with 64 Frames. Beyond the 32-frame setting, we further evaluate DynVid with 64 sampled frames, which provide denser temporal coverage and capture finer-grained temporal details, while also introducing more visual tokens and higher redundancy. Table 7 reports the detailed results on eight benchmarks for LLaVA-OneVision and Qwen2.5-VL. DynVid achieves higher average performance than FlashVID across all four retention ratios on both models, with the advantage generally becoming more pronounced under aggressive compression. At $\bar{R}$=5%, DynVid improves the average score from 53.8 to 55.4 on LLaVA-OneVision and from 55.7 to 56.9 on Qwen2.5-VL. Moreover, on LLaVA-OneVision, DynVid at $\bar{R}$=20% and 10% achieves 101.2% and 100.4% relative accuracy, respectively, slightly surpassing the uncompressed 64-frame baseline. These results demonstrate that DynVid remains effective with denser video inputs and preserves informative content under increasingly constrained token budgets.

E Hyperparameter Sensitivity

Anchor Aggregation Weight γ. We examine the sensitivity to the anchor aggregation weight γ in Eq. 14, which balances the original anchor representation and the aggregated non-anchor information during soft fusion. We sweep $\gamma \in \{0.1, 0.25, 0.5, 0.75, 0.9, 1.0\}$ at $\bar{R}$=10%. At γ=1, the anchor retains only its own representation without absorbing any non-anchor context; as γ decreases, non-anchor tokens contribute more to the fused anchor representation, enabling richer information preservation at the cost of potentially diluting the anchor's original semantics. As shown in Figure 4, the best Avg Score is achieved at γ=0.1 on LLaVA-OneVision and γ=0.25 on Qwen2.5-VL, with performance declining slightly as γ increases, indicating that incorporating non-anchor context effectively recovers information that would otherwise be lost after compression. We uniformly set γ=0.25 by default.

Dynamic Segmentation Threshold τ_s. We analyze the sensitivity to the segmentation threshold τ_s, which controls the granularity of temporal segmentation: consecutive frames are merged into the same segment when their cosine similarity exceeds τ_s. We sweep $\tau_s \in \{0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7\}$ at $\bar{R}$=10%. A lower τ_s produces fewer, longer segments

Table 7: Detailed results with 64 sampled frames on eight benchmarks. **Bold** denotes the best performance.

Method	Retention $\bar{R}$	VideoMME	EgoSchema	LongVideo Bench	MVBench	MLVU Test	MLVU Dev	LVBench	Temp Compass	Avg. Score	Avg. Rel.(%)
LLaVA-OneVision (64 frames)											
Vanilla	100%	58.8	60.2	57.9	56.8	47.8	64.9	39.3	63.5	56.2	100.0
FlashVID	20%	59.0	59.8	57.4	58.3	47.4	64.9	**40.6**	**65.2**	56.6	100.7
DynVid		**59.2**	**60.4**	**58.4**	**58.7**	**49.2**	**66.1**	38.4	64.6	**56.9**	**101.2**
FlashVID	10%	58.4	59.9	56.7	57.3	46.5	64.5	37.1	**64.9**	55.7	99.1
DynVid		**58.9**	**60.6**	**58.1**	**57.7**	**47.4**	**65.2**	**38.7**	64.6	**56.4**	**100.4**
FlashVID	7.5%	57.2	59.5	56.5	**57.1**	45.1	63.8	37.2	**64.9**	55.2	98.2
DynVid		**57.7**	**60.4**	**58.5**	57.0	**48.2**	**64.3**	**38.7**	64.1	**56.1**	**99.8**
FlashVID	5%	55.9	58.7	54.8	56.4	43.1	61.8	36.0	**64.0**	53.8	95.7
DynVid		**57.6**	**60.6**	**56.8**	**56.8**	**45.6**	**64.4**	**37.8**	63.4	**55.4**	**98.6**
Qwen2.5-VL (64 frames)											
Vanilla	100%	63.5	58.6	60.4	69.0	46.6	63.9	38.9	74.1	59.4	100.0
FlashVID	20%	61.8	57.7	**58.6**	68.2	45.4	**62.2**	37.9	73.0	58.1	97.8
DynVid		**62.8**	**58.2**	**58.6**	**68.6**	**46.6**	61.9	**38.0**	**74.0**	**58.6**	**98.7**
FlashVID	10%	60.7	57.3	57.2	**67.7**	**44.7**	60.5	36.7	**73.4**	57.3	96.5
DynVid		**61.4**	**57.6**	**58.3**	67.4	**44.7**	**60.6**	**38.4**	72.7	**57.6**	**97.0**
FlashVID	7.5%	60.2	56.7	57.4	**67.4**	44.0	60.1	36.5	72.2	56.8	95.6
DynVid		**61.2**	**57.3**	**58.1**	67.0	**46.3**	**60.5**	**37.3**	**72.5**	**57.5**	**96.8**
FlashVID	5%	59.1	56.2	56.5	**66.5**	42.6	58.6	35.9	70.0	55.7	93.8
DynVid		**60.1**	**56.6**	**57.3**	66.4	**45.6**	**59.8**	**38.0**	**71.6**	**56.9**	**95.8**

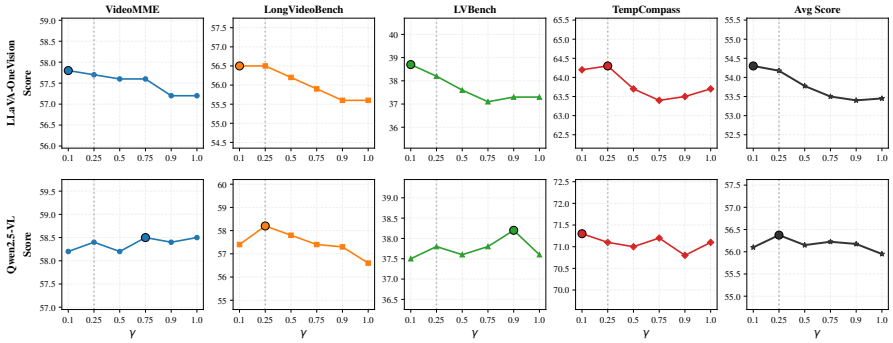


Figure 4: Sensitivity analysis of anchor aggregation weight γ on four benchmarks at $\bar{R}$=10%. LLaVA-OneVision peaks at γ=0.1 and Qwen2.5-VL at γ=0.25.

that group semantically similar frames together, while a higher τ_s yields finer-grained segments that better capture scene transitions but may over-fragment static content. As shown in Figure 5, the best Avg Score is achieved at τ_s=0.6 for LLaVA-OneVision (54.3) and τ_s=0.4 for Qwen2.5-VL (56.4). Beyond these optima, performance declines on both models (*e.g.*, Avg drops from 54.3 to 53.7 on LLaVA-OneVision at τ_s=0.65), suggesting that over-segmentation splits semantically coherent scenes into isolated fragments, disrupting temporal continuity and reducing the effectiveness of segment-level budget allocation. The optimal τ_s differs across models, which we attribute to differences in vision encoder architectures that produce distinct inter-frame similarity distributions, requiring different thresholds to achieve the best segmentation granularity. In practice, τ_s can be selected via a small validation sweep with minimal cost, as the method is training-free.

Expansion factor f_e. We study the effect of the expansion factor f_e that controls how many visual tokens the before-LLM compression stage retains as candidates for subsequent inner-LLM pruning ($B = f_e \cdot \bar{R}$). A larger f_e retains more candidate tokens entering the LLM but requires the inner-LLM stage to prune a larger fraction to reach the target $\bar{R}$; a smaller f_e concentrates compression before the LLM, reducing the tokens entering the LLM and increasing the risk of discarding task-relevant content without query guidance. We sweep f_e from 1.0 to 1.35 at $\bar{R} = 10\%$ on four benchmarks. As shown in Figure 6, both LLaVA-OneVision and Qwen2.5-VL peak at $f_e = 1.25$ (54.2 and 56.4 average score, respectively). When $f_e = 1.0$, all compression occurs before the LLM

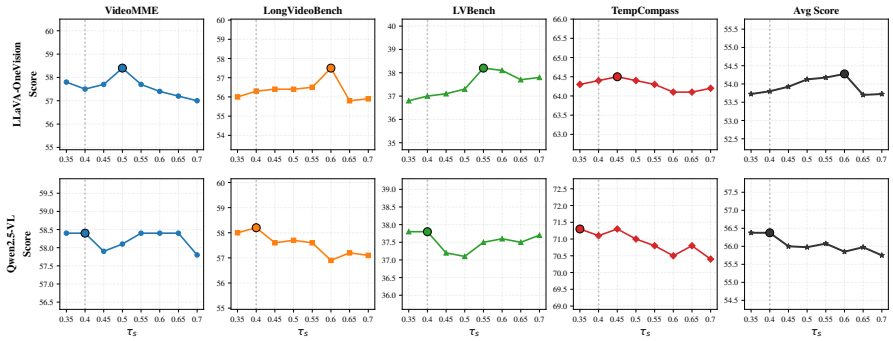


Figure 5: Sensitivity analysis of dynamic segmentation threshold τ_s on four benchmarks at $\bar{R}$=10%. LLaVA-OneVision peaks at τ_s=0.6 and Qwen2.5-VL at τ_s=0.4.

without query guidance, risking over-pruning of task-relevant tokens and leading to a noticeable performance drop on both models. Conversely, when f_e exceeds 1.25, the inner-LLM stage must prune a larger fraction to maintain the same $\bar{R}$, increasing the risk of discarding query-relevant tokens. The consistent optimum at $f_e = 1.25$ across both architectures balances these two factors: retaining sufficient tokens before the LLM to preserve broad content coverage, while keeping the inner-LLM pruning moderate enough to avoid discarding excessive task-relevant information. We fix this value across all benchmarks and VLMs without further tuning.

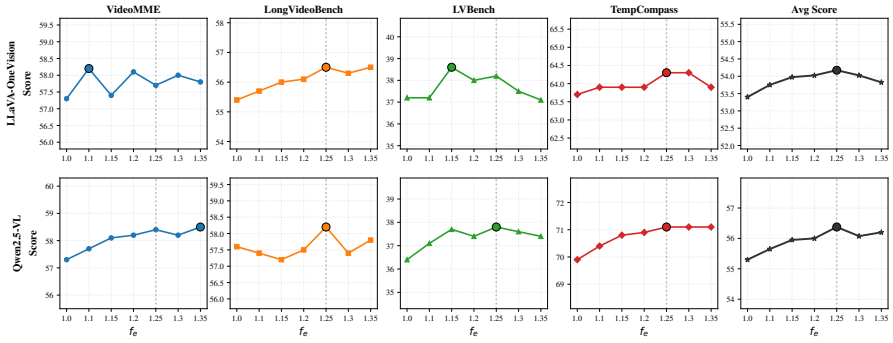


Figure 6: Sensitivity analysis of expansion factor f_e on four benchmarks at $\bar{R}$=10%. Both LLaVA-OneVision (top) and Qwen2.5-VL (bottom) peak at f_e=1.25.

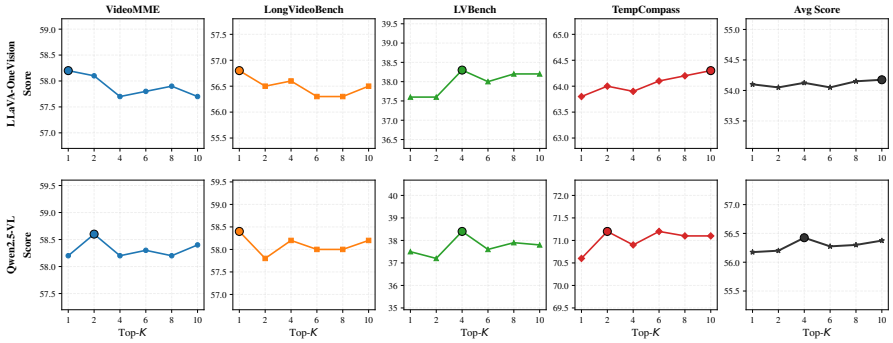


Figure 7: Sensitivity analysis of the soft fusion top-K on four benchmarks at $\bar{R}$=10%. Performance is largely insensitive to K on both LLaVA-OneVision (top) and Qwen2.5-VL (bottom).

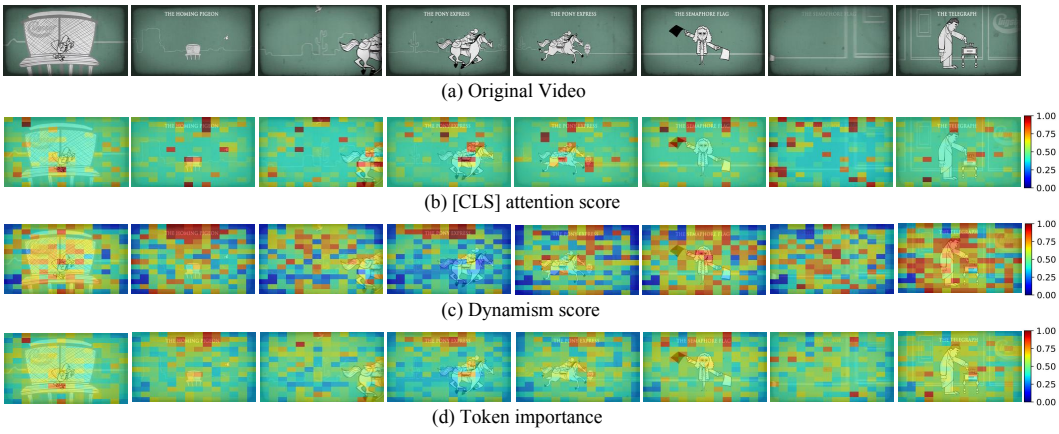
Soft Fusion Top-K. During soft fusion, each non-anchor token is assigned to its top-K most similar anchors with softmax weights (Eq. 13), aggregating useful information from non-anchor tokens into the retained anchors. We vary $K \in \{1, 2, 4, 6, 8, 10\}$ at $\bar{R}$=10%. As shown in Figure 7, performance is largely insensitive to K on both models, with average scores varying by less than 0.2 points. However, K=1 yields slightly lower performance, as a non-anchor token may share semantic similarity with multiple anchors, and restricting assignment to a single anchor discards this shared information. A moderate K allows the non-anchor context to be distributed across several relevant anchors via softmax weighting, better preserving information of non-anchor tokens. We set K=10 consistently across all benchmarks and VLMs without further tuning.

F VISUALIZATION

Qualitative Visualization of Token Dynamism. We visualize the token importance computation in Figures 8–13 across diverse video types (animation, whiteboard lecture, and surveillance footage). The `[CLS]` attention score (row b) captures semantic saliency, assigning high scores to visually prominent regions without explicitly accounting for temporal change. The dynamism score (row c) provides a complementary temporal signal, highlighting regions undergoing cross-frame change (*e.g.*, a galloping horse, a hand actively drawing, moving pedestrians) while suppressing static content even if visually complex. The fused token importance (row d) combines both signals, prioritizing tokens that are simultaneously salient and temporally informative. For example, in whiteboard videos the actively drawn region is emphasized while the static diagram skeleton is down-weighted; in surveillance footage, moving pedestrians dominate over the stationary background. This dual-signal fusion enables DynVid to allocate limited token budgets to the most informative visual content, maximizing information coverage for subsequent query-guided pruning.

Visualization of FL-based Anchor Selection. We compare the distribution of anchor tokens selected by our Facility Location (FL) method and the naive top-k baseline in Figures 14–17. Top-k selection greedily picks tokens with the highest individual importance scores, which causes severe semantic clustering: selected tokens concentrate in a few high-score regions while leaving other informative areas entirely unrepresented (*e.g.*, in Example 2 on Qwen2.5-VL, top-k assigns 158/720 tokens to a single scene-transition frame while allocating only 1 token to another). In contrast, our FL-based selection maximizes content coverage by ensuring that every token is well-represented by at least one similar anchor, producing a dispersed and balanced distribution. This ensures that even lower-importance but semantically distinct regions retain representative anchors, preserving broader visual context after compression.

Case Study: Correct and Incorrect Predictions. We present qualitative QA examples in Figures 18–25, showing cases where DynVid corrects the vanilla model's errors and cases where it introduces new mistakes. In the success cases, DynVid helps the model focus on temporally relevant content that the vanilla model overlooks. For instance, on LLaVA-OneVision (Example 1), the question asks about the communication method listed before "Semaphore"; DynVid correctly identifies "Pony express" by preserving tokens from the key transitional frames, whereas the vanilla model confuses the ordering. Similarly, on Qwen2.5-VL (Example 2), DynVid retains key frames depicting a lunar impact scene that the vanilla model misinterprets as a meteorite attack. In the failure cases, compression may discard fine-grained visual details required for certain questions. On LLaVA-OneVision (Example 3), answering "what color is the cell membrane" demands pixel-level color discrimination that may be lost after token reduction. On Qwen2.5-VL (Example 4), DynVid may discard tokens carrying critical textual and discriminative information needed to distinguish between answer choices, leading to an incorrect prediction. These failure modes suggest that questions requiring fine-grained static details or dense text reading remain challenging for compression-based methods, pointing to potential future directions such as query-adaptive budget allocation prior to LLM processing.



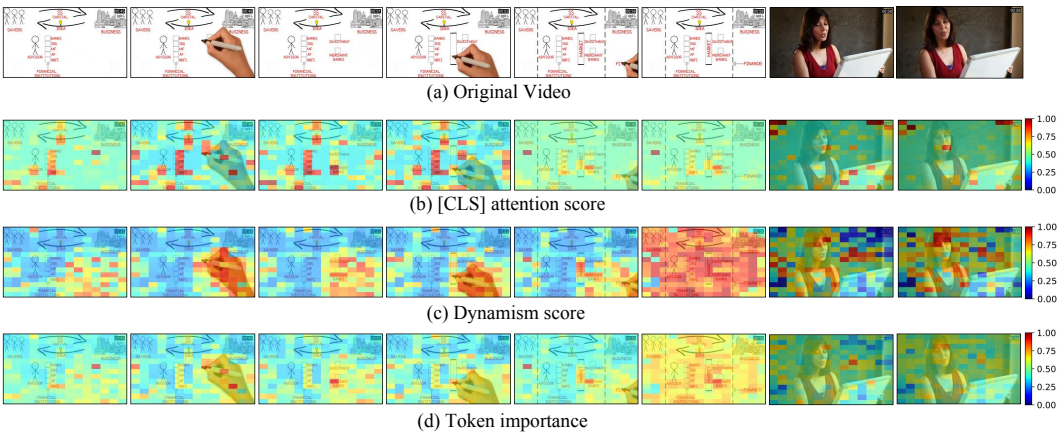
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 8: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **LLaVA-OneVision** (Example 1). The fused importance effectively highlights dynamic and salient regions while suppressing static backgrounds.



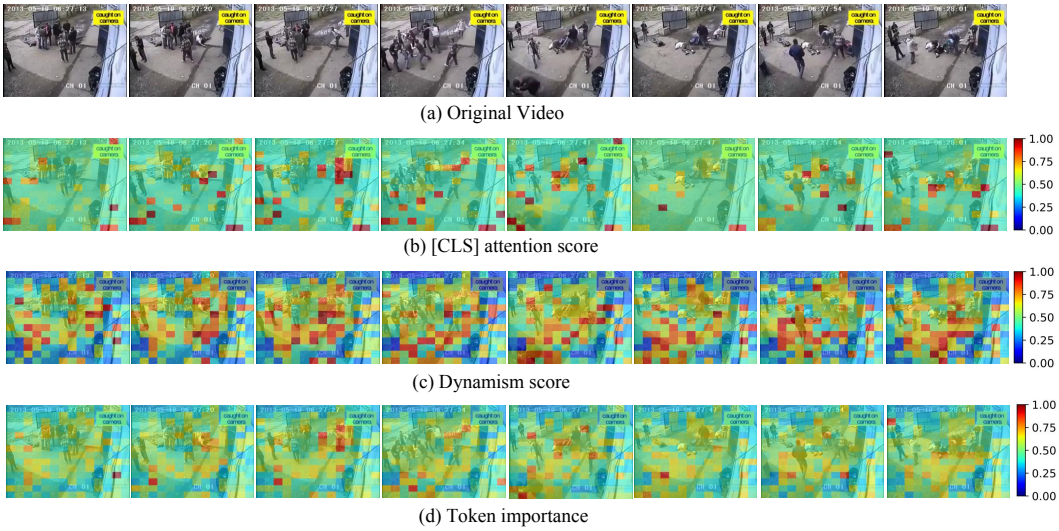
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 9: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **LLaVA-OneVision** (Example 2).



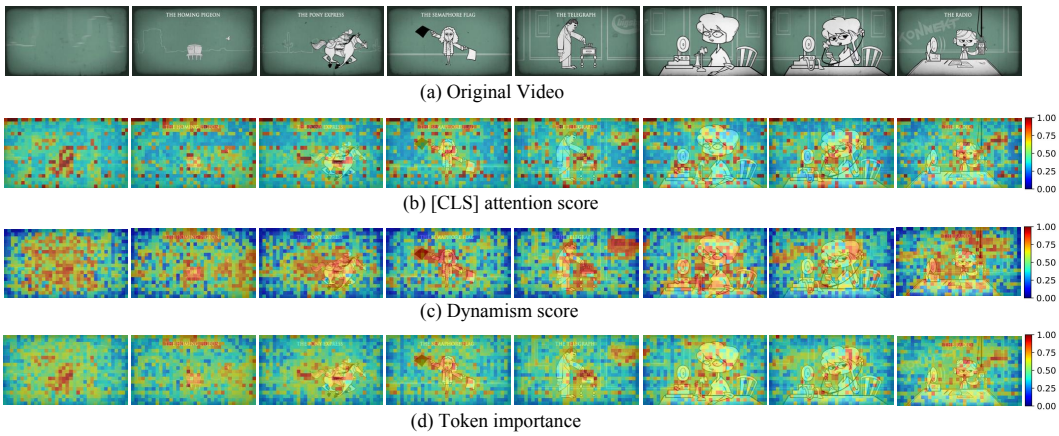
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 10: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **LLaVA-OneVision** (Example 3).



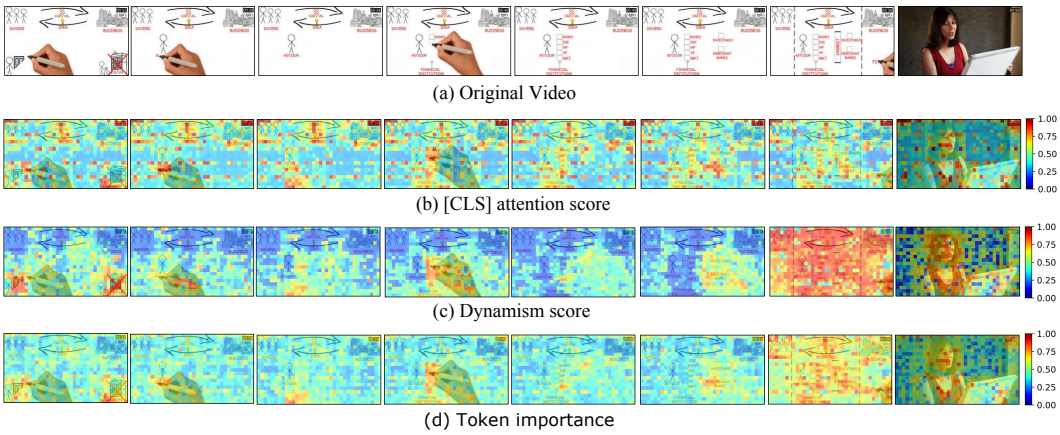
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 11: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **Qwen2.5-VL** (Example 1).



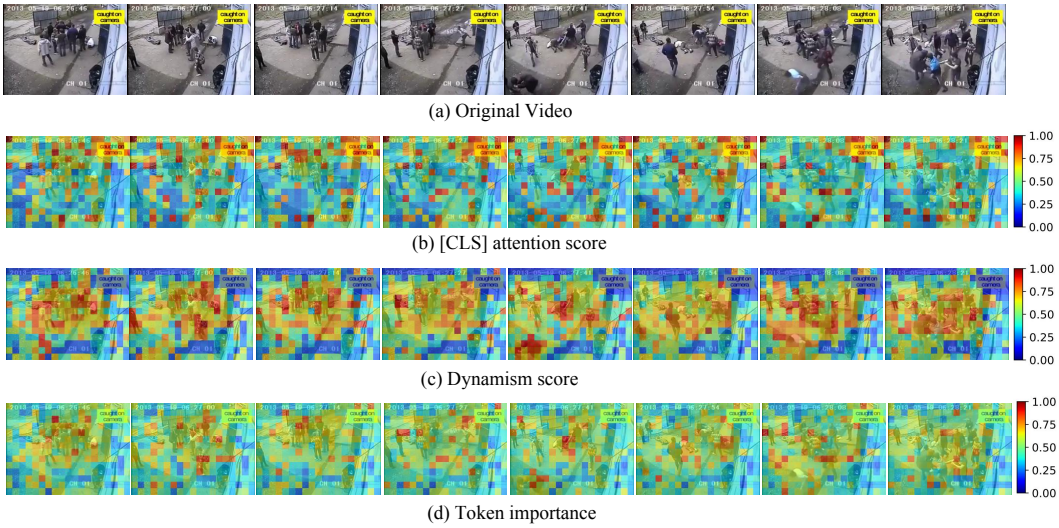
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 12: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **Qwen2.5-VL** (Example 2).



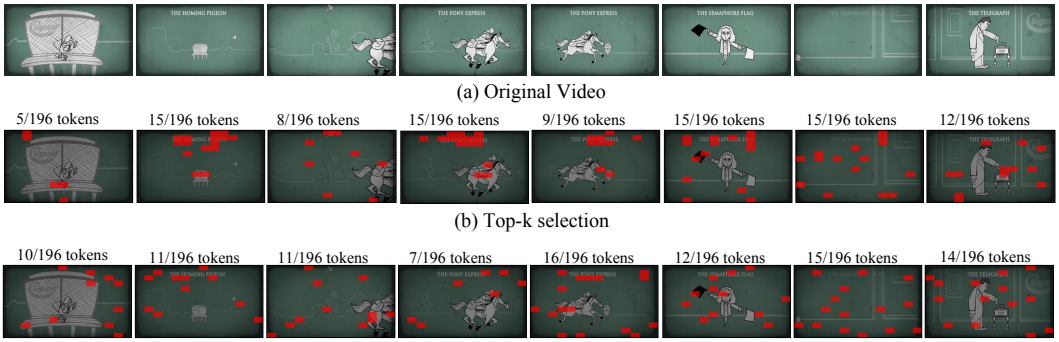
(a) Original Video

(b) [CLS] attention score

(c) Dynamism score

(d) Token importance

Figure 13: Visualization of `[CLS]` attention score, dynamism score and the final token importance on **Qwen2.5-VL** (Example 3).

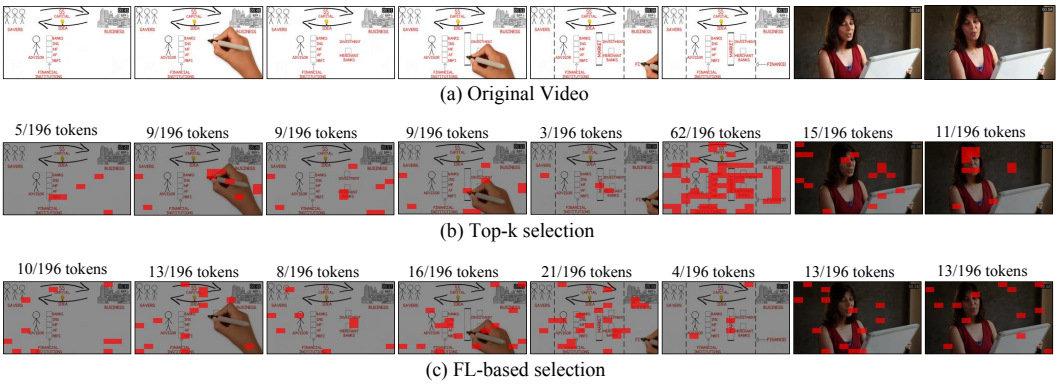


(a) Original Video

(b) Top-k selection

(c) FL-based selection

Figure 14: Visualization comparison of anchor tokens selected by our Facility Location method and top-k method on **LLaVA-OneVision** (Example 1). FL achieves broader spatial coverage while top-k clusters tokens in localized high-score regions.



(a) Original Video

(b) Top-k selection

(c) FL-based selection

Figure 15: Visualization comparison of anchor tokens selected by our Facility Location method and top-k method on **LLaVA-OneVision** (Example 2).



(a) Original Video



(b) Top-k selection

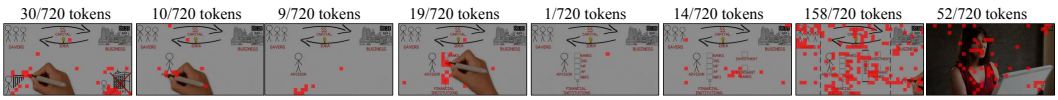


(c) FL-based selection

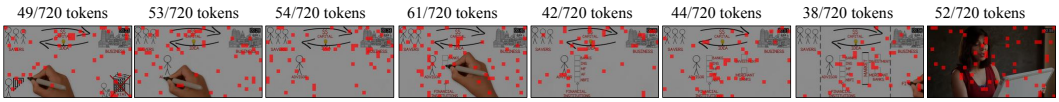
Figure 16: Visualization comparison of anchor tokens selected by our Facility Location method and top-k method on **Qwen2.5-VL** (Example 1).



(a) Original Video



(b) Top-k selection



(c) FL-based selection

Figure 17: Visualization comparison of anchor tokens selected by our Facility Location method and top-k method on **Qwen2.5-VL** (Example 2).



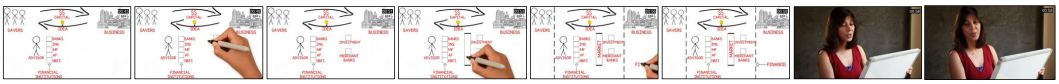
Question:

What kind of communication is listed before Semaphore? A. Telephone. B. Homing pigeon. C. Telegraph. D. Pony express.

LLaVA-OneVision *C. Telegraph.* ✗

LLaVA-OneVision w/ DynVid *D. Pony express.* ✓

Figure 18: Qualitative example where DynVid corrects the vanilla model's prediction on **LLaVA-OneVision** (Example 1).



Question:

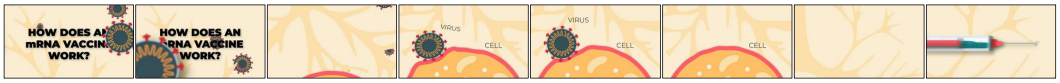
What is the function of the stopwatch in the upper right corner of the video?

A. To indicate the total duration of this video. B. The video author made the annotations in order to explain clearly what finance is within one minute. C. To display the current time when the narrator is sharing financial knowledge. D. It is a countdown timer.

LLaVA-OneVision *C. To display the current time when the narrator is sharing financial knowledge.* ✗

LLaVA-OneVision w/ DynVid *B. The video author made the annotations in order to explain clearly what finance is within one minute.* ✓

Figure 19: Qualitative example where DynVid corrects the vanilla model's prediction on **LLaVA-OneVision** (Example 2).



Question:

What color is the cell membrane of the virus-invaded cell in the video? A. Black. B. Yellow. C. Red. D. Blue.

LLaVA-OneVision *C. Red.* ✓

LLaVA-OneVision w/ DynVid *B. Yellow.* ✗

Figure 20: Qualitative example where DynVid introduces an incorrect prediction on **LLaVA-OneVision** (Example 3).



Question:

Which historical figure is visually depicted in a framed portrait in the video?

A. Abraham Lincoln. B. George Washington. C. John F. Kennedy. D. Franklin D. Roosevelt.

LLaVA-OneVision *D. Franklin D. Roosevelt.* ✓

LLaVA-OneVision w/ DynVid *A. Abraham Lincoln.* ✗

Figure 21: Qualitative example where DynVid introduces an incorrect prediction on **LLaVA-OneVision** (Example 4).



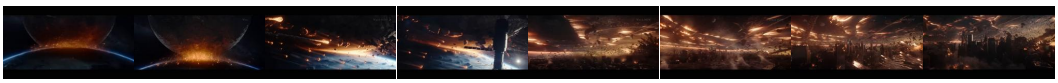
Question:

Which color of clothing is worn by the first person selling bananas in the video? A. Blue. B. Purple. C. Black. D. Green.

Qwen2.5-VL *C. Black.* ✗

Qwen2.5-VL w/ DynVid *A. Blue.* ✓

Figure 22: Qualitative example where DynVid corrects the vanilla model's prediction on **Qwen2.5-VL** (Example 1).



Question:

What is the most likely cause of the disaster in the video? A. Warfare. B. Meteorite attack. C. Lunar impact. D. Alien attack.

Qwen2.5-VL	*B. Meteorite attack.*	✗
Qwen2.5-VL w/ DynVid	*C. Lunar impact.*	✓

Figure 23: Qualitative example where DynVid corrects the vanilla model's prediction on **Qwen2.5-VL** (Example 2).



Question:

According to the video, which of the following is considered the earliest stage of human evolution?

A. Ramapithecus. B. Dryopithecus. C. Ardipithecus Ramidus. D. Homo Sapiens.

Qwen2.5-VL	*B. Dryopithecus.*	✓
Qwen2.5-VL w/ DynVid	*C. Ardipithecus Ramidus.*	✗

Figure 24: Qualitative example where DynVid introduces an incorrect prediction on **Qwen2.5-VL** (Example 3).



Question:

What is special about the celebration in New York according to the video?

A. Hosting large parades. B. Dressing in green and dyeing the river to green. C. Drinking a lot. D. Planting shamrocks.

Qwen2.5-VL	*A. Hosting large parades.*	✓
Qwen2.5-VL w/ DynVid	*B. Dressing in green and dyeing the river to green.*	✗

Figure 25: Qualitative example where DynVid introduces an incorrect prediction on **Qwen2.5-VL** (Example 4).