AVTC: AGGRESSIVE VISUAL TOKEN COMPRESSION VIA SET-AWARE SELECTION FOR VIDEO LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token compression offers a practical way to reduce the inference cost of video large language models (Video-LLMs), but preserving model performance under aggressive compression remains challenging. Existing methods often select tokens by individual importance or prune them based on pairwise similarity, which can lead to redundant selections and incomplete coverage of the original visual content under tight budgets. To address this limitation, we introduce AVTC, a training-free framework that performs set-aware selection by evaluating how well the selected tokens jointly reconstruct the original visual features. Specifically, AVTC scores each candidate by how much its addition reduces the total importance-weighted projection error of all original tokens onto the subspace spanned by the selected anchors. This criterion favors tokens that contribute important information not yet captured by the current selection. Experiments on three Video-LLMs and four benchmarks show that AVTC achieves the highest average accuracy among the compared methods at layer-average token retention ratios of 2%, 5%, and 10%. On LLaVA-OneVision-7B, AVTC retains **94.6%** of the uncompressed model's average accuracy at just **2%** token retention. At **10%** retention, it achieves **100.2%** of the uncompressed model's average accuracy while delivering a **6.6**× prefill speedup and a **2.0**× speedup in time to first token (TTFT). Code is available at https://anonymous.4open.science/r/AVTC-8F87.

1 INTRODUCTION

Recent advances in Video Large Language Models (Video-LLMs) (Lin et al., 2024; Li et al., 2025; Zhang et al., 2025b; Bai et al., 2025b;a) have demonstrated strong video understanding capabilities (Li et al., 2024; Fu et al., 2025; Wu et al., 2024; Mangalam et al., 2023) by integrating visual perception with language reasoning. However, videos are typically encoded into far more visual tokens than static images, and processing such long token sequences incurs substantial inference costs, as the attention mechanism scales quadratically with sequence length. This overhead has become a major bottleneck for the practical deployment of Video-LLMs. To address this challenge, visual token compression has emerged as an active research direction (Chen et al., 2024; Yang et al., 2025; Shen et al., 2025; Cho et al., 2026). These methods shorten visual token sequences through pruning or merging, aiming to accelerate inference while maintaining model performance.

Existing visual token compression methods (Chen et al., 2024; Yang et al., 2025; Fan et al., 2026; Shen et al., 2025; Shao et al., 2025a; Wang et al., 2026) typically rely on two types of local signals to guide token pruning or merging. The first is attention-based estimation of individual token importance, where tokens receiving higher attention are considered more important and thus retained. The second measures inter-token redundancy using pairwise feature similarity, such as cosine similarity, and prunes or merges tokens with highly similar features. However, these local criteria do not directly assess how well the retained subset jointly represents the original token set and ignore the interdependencies among individual selection decisions. Individually important tokens may convey overlapping information, while pairwise similarity can also fail to identify redundant tokens because several selected tokens may jointly represent a token that is dissimilar to each of them. Under a tight token budget, redundant selections leave fewer slots for complementary information, making

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

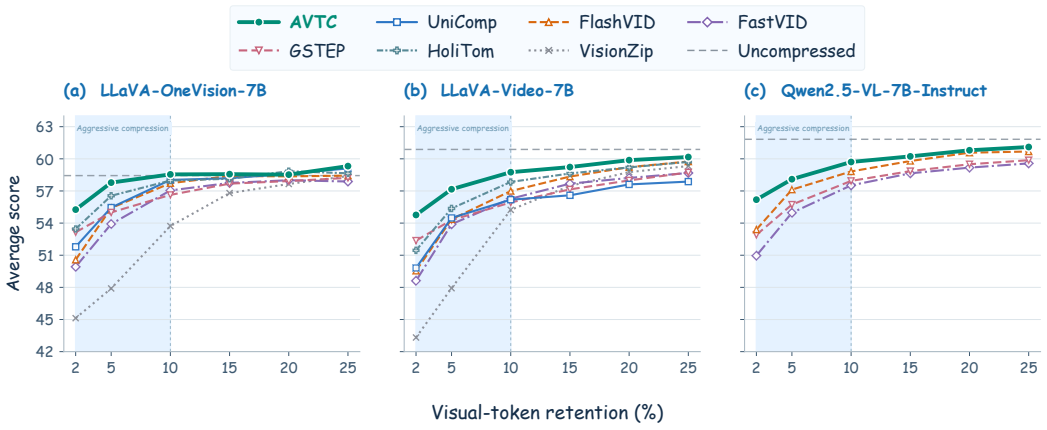

Figure 1: **Performance comparison across visual token retention ratios on three Video-LLM backbones.** The y-axis reports the average score over Video-MME, LongVideoBench, MVBench, and EgoSchema. AVTC's performance advantage becomes increasingly pronounced as the retention ratio decreases (shaded region).

these limitations particularly consequential. As illustrated in Figure 1, most existing methods suffer substantial performance degradation when the token retention ratio drops below 10%.

To better preserve visual information under aggressive compression, we propose AVTC, an **A**ggressive **V**isual **T**oken **C**ompression framework that performs set-aware token selection without additional training. AVTC seeks a compact set of tokens, termed anchors, that jointly reconstruct the original visual features. We formulate this goal as minimizing the total importance-weighted projection error of all original tokens onto the subspace spanned by the selected anchors. AVTC greedily selects the candidate whose addition most reduces this total error. Importance weighting prioritizes informative content, while minimizing the total projection error promotes coverage of the original visual information. To reduce the overhead of this greedy selection, AVTC employs segment-wise parallel evaluation. After selection, AVTC selectively merges unselected tokens into anchors to retain useful information and calibrates attention to mitigate the bias introduced by merging. Text-guided compression within the LLM then further reduces computation.

We implement AVTC on three representative Video-LLMs, including LLaVA-OneVision-7B (Li et al., 2025), LLaVA-Video-7B (Zhang et al., 2025b), and Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), and evaluate its effectiveness across four video understanding benchmarks. As shown in Table 1, AVTC outperforms the compared methods in average accuracy across all three backbones at layer-average token retention ratios of 2%, 5%, and 10%. Notably, the average accuracy improvement over the strongest baseline grows as retention decreases, reaching 1.80–2.79 points at 2% retention (Figure 1). In addition, at 10% retention on LLaVA-OneVision-7B, AVTC slightly exceeds the uncompressed model's average accuracy while achieving $6.6\times$ and $2.0\times$ speedups in total prefill and time to first token (TTFT), respectively.

Our main contributions are threefold:

- We identify a key limitation in commonly used visual token selection criteria, which do not directly assess how each selection improves the retained set's joint reconstruction of the original visual features. Under tight token budgets, this can leave useful visual information poorly represented even when the retained tokens are individually important or diverse.
- We propose AVTC, an **A**ggressive **V**isual **T**oken **C**ompression framework for Video-LLMs. Its set-aware selection minimizes the total importance-weighted reconstruction error of the original visual features, prioritizing informative content and promoting complementary selections under tight token budgets.
- We implement AVTC on three Video-LLM backbones and evaluate it across four benchmarks, showing consistent gains in average accuracy over competing methods under aggressive compression. Efficiency measurements and extensive ablation studies further demonstrate practical inference speedups and the effectiveness of AVTC's key components.

2 Related Work

2.1 Video Large Language Models

Video Large Language Models (Video-LLMs) extend image-based multimodal LLMs to the temporal domain. They typically sample frames from a video, encode them with a visual encoder, and feed the projected visual features together with text into an LLM (Lin et al., 2024; Li et al., 2025). Early efforts such as Video-ChatGPT (Maaz et al., 2024) and VideoChat (Li et al., 2023) bridge video representations to an LLM via lightweight adaptation modules. Subsequent work strengthens spatio-temporal understanding through unified image–video training, higher-resolution visual inputs, and denser frame sampling. LLaVA-OneVision (Li et al., 2025) unifies image, multi-image, and video understanding within a single model, and LLaVA-Video (Zhang et al., 2025b) scales up video instruction tuning with synthetic data. Qwen2.5-VL (Bai et al., 2025b) supports native dynamic resolution and dynamic frame-rate sampling for video inputs, and Qwen3-VL (Bai et al., 2025a) further advances long-context video modeling with enhanced interleaved MRoPE and text-based temporal alignment. Across these models, richer visual representations come with a rapidly growing number of visual tokens, leading to substantial inference overhead. This makes effective visual token compression increasingly important for the practical deployment of Video-LLMs.

2.2 Visual Token Compression for Video-LLMs

Existing visual token compression methods typically use attention scores to identify important tokens (Chen et al., 2024; Yang et al., 2025) or feature similarity to guide token pruning and merging (Bolya et al., 2023; Dhouib et al., 2025; Tao et al., 2025). Video-specific methods further extend these principles to the temporal dimension to cope with strong cross-frame redundancy. FastVID (Shen et al., 2025) applies density-based pruning within temporal segments, HoliTom (Shao et al., 2025a) combines temporal segmentation with outer- and inner-LLM merging, FlashVID (Fan et al., 2026) performs attention- and diversity-based selection followed by spatio-temporal tree merging, and UniComp (Yuan et al., 2026) estimates the informational uniqueness of tokens to guide frame grouping and spatial compression. Despite their effectiveness, these methods still rely on local signals such as token-wise importance, local density, pairwise diversity, or cluster structure. These selection rules do not directly assess how adding a token improves the retained set's joint reconstruction of the original visual features.

Recent methods also formulate token selection using explicit coverage or reconstruction criteria. FLoC (Cho et al., 2026) adopts a facility location objective that maximizes the summed similarity between original tokens and their best-matching selected tokens. SCOPE (Deng et al., 2025) combines marginal coverage gains with candidate saliency to guide iterative selection. Both methods measure coverage through individual token matches, without assessing how multiple selected tokens jointly reconstruct the original visual features. SPARE (Lee et al., 2026) formulates pruning as column subset selection and combines projection residuals with anti-relevance guidance for token selection. ResPrune (Li et al., 2026) similarly selects tokens by residual energy conditioned on textual relevance. Both methods score each candidate by its own residual, i.e., the component of the candidate itself that is not explained by the tokens already selected. In contrast, AVTC selects the token that most reduces the total importance-weighted projection error of the original token set onto the subspace spanned by the retained tokens.

3 Method

3.1 Problem Formulation and Design Overview

Video large language models (Video-LLMs) typically use a vision encoder to extract visual features and a projector to map them into the embedding space of a large language model (LLM). The resulting visual tokens are then concatenated with text tokens and fed into the LLM for downstream reasoning. Formally, given a video clip, we denote the projected visual token sequence before compression by

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}, \tag{1}$$

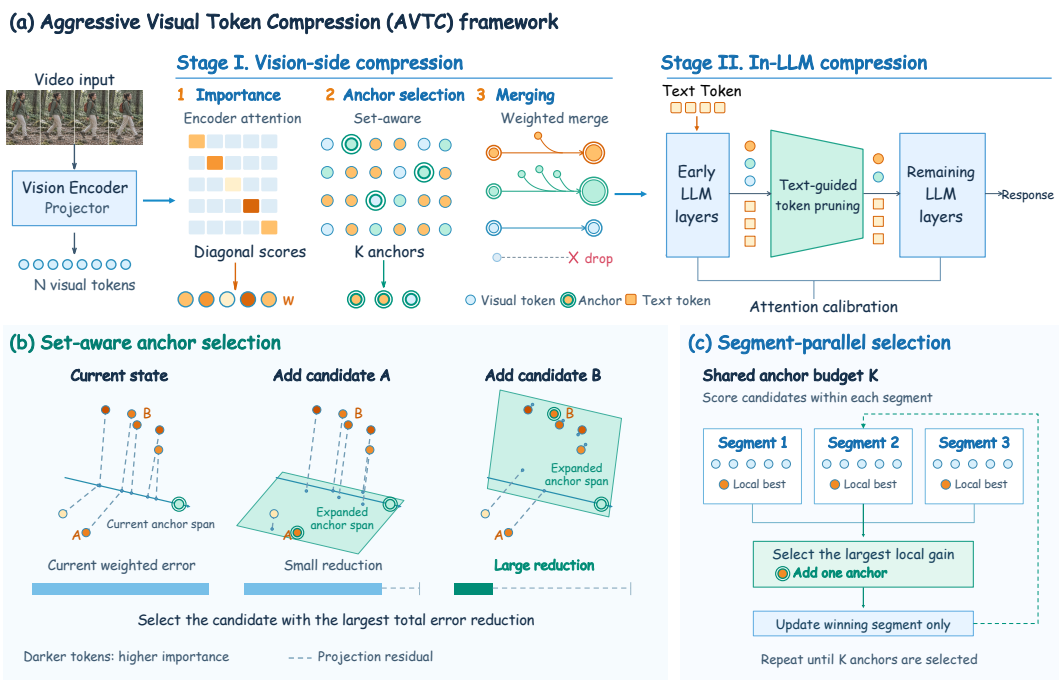

Figure 2: **Overview of the AVTC framework.** (a) The overall pipeline of AVTC, which compresses visual tokens in two stages: set-aware compression before the LLM and text-guided compression within the LLM. (b) Set-aware anchor selection. AVTC maintains a feature subspace spanned by the selected anchors and iteratively adds the token that most reduces the importance-weighted projection error of the original token set. (c) Segment-wise parallel selection. The token sequence is partitioned into temporal segments, and candidate evaluation is performed in parallel within each segment.

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the i-th visual token, N is the total number of visual tokens before compression, and d is the LLM embedding dimension. The sequence length N depends on the number of sampled frames and the backbone's visual tokenization scheme.

We consider visual token compression both before and within the LLM, aiming to reduce inference cost while maintaining video understanding performance. Before the LLM, compression maps $\mathbf{X}$ to a compact sequence:

$$\mathbf{X}' = f(\mathbf{X}; K) \in \mathbb{R}^{K \times d}, \tag{2}$$

where K is the visual token budget at the LLM input. At an intermediate LLM layer ℓ_c, a second compression stage retains a fraction ρ_c of the visual tokens present at that layer, further shortening the sequence for subsequent layers. To account for both stages, we define the layer-average token retention ratio as

$$\rho = \frac{1}{LN} \sum_{\ell=1}^{L} N_\ell, \tag{3}$$

where L is the number of LLM layers and N_ℓ is the number of visual tokens processed by layer ℓ. This ratio measures the average visual token count across LLM layers relative to the uncompressed sequence length N.

Figure 2 provides an overview of AVTC framework. AVTC first estimates token importance from attention in the vision encoder (Sec. 3.2). It then performs set-aware anchor selection by evaluating how much each candidate's addition reduces the total importance-weighted projection error of the original tokens (Sec. 3.3). After selection, unselected tokens are selectively merged into temporally nearby anchors (Sec. 3.4). Within the LLM, we calibrate attention to account for the number of original tokens represented by each anchor, while text-guided compression further shortens the visual sequence (Sec. 3.5).

3.2 VISUAL-TOKEN IMPORTANCE ESTIMATION

AVTC first estimates token-wise importance scores to weight the reconstruction errors of the original visual tokens during anchor selection. We use the diagonal entries of existing vision-encoder attention matrices as a proxy for token importance, averaging them across heads in the final layer. This requires neither an additional encoder forward pass nor a dedicated `[CLS]` token, allowing the estimator to be used with encoders such as SigLIP (Zhai et al., 2023) adopted in LLaVA-OneVision.

Formally, let $\mathbf{H}^{(i)} \in \mathbb{R}^{M\times d_{\mathrm{v}}}$ denote the token features of the i-th frame at the input of the last Transformer layer of the vision encoder, where M is the number of tokens in that frame and d_{v} is the vision encoder's feature dimension. For the h-th attention head, the self-attention matrix is computed as

$$\mathbf{A}_h^{(i)} = \mathrm{softmax}\left(\frac{\mathbf{Q}_h^{(i)}\big(\mathbf{K}_h^{(i)}\big)^\top}{\sqrt{d_h}}\right) \in \mathbb{R}^{M\times M}, \tag{4}$$

where $\mathbf{Q}_h^{(i)} = \mathbf{H}^{(i)}\mathbf{W}_h^Q$ and $\mathbf{K}_h^{(i)} = \mathbf{H}^{(i)}\mathbf{W}_h^K$ are the query and key projections, and d_h is the head dimension. The importance score of the j-th token in the i-th frame is then defined as the diagonal entry of the attention matrix averaged over all H heads:

$$\tilde{w}_j^{(i)} = \frac{1}{H}\sum_{h=1}^{H}\big[\mathbf{A}_h^{(i)}\big]_{jj}. \tag{5}$$

We apply min-max normalization within each frame to obtain $w_j^{(i)} \in [0,1]$. The score maps are aligned with the projected token sequence $\mathbf{X}$ through the corresponding spatial pooling or interpolation. The aligned scores are indexed as $\{w_i\}_{i=1}^N$ and used as importance weights in subsequent anchor selection and token merging.

3.3 ANCHOR TOKEN SELECTION

Formulation. Given the projected visual tokens $\mathbf{X}$ and their importance weights $\{w_i\}_{i=1}^N$, we formulate anchor selection as a weighted reconstruction problem. For selection, we use the ℓ_2-normalized features $\hat{\mathbf{x}}_i = \mathbf{x}_i/\|\mathbf{x}_i\|_2$ to prevent feature magnitudes from dominating the reconstruction error. The original features $\mathbf{x}_i$ are retained for token merging. We seek an anchor index set $\mathcal{A} \subseteq \{1,\dots,N\}$ with $|\mathcal{A}| = K$ that minimizes the total importance-weighted projection error:

$$\mathcal{A}^* = \operatorname*{arg\,min}_{|\mathcal{A}|=K} \mathcal{E}(\mathcal{A}), \quad \text{where} \quad \mathcal{E}(\mathcal{A}) = \sum_{i=1}^{N} w_i \left\|\hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}}\,\hat{\mathbf{x}}_i\right\|_2^2, \tag{6}$$

where $\mathbf{P}_{\mathcal{A}}$ is the orthogonal projector onto $\mathrm{span}\{\hat{\mathbf{x}}_k : k \in \mathcal{A}\}$. This objective evaluates how well the anchors jointly reconstruct the normalized features of all original tokens, with each reconstruction error weighted by w_i. Adding a token that can be exactly reconstructed as a linear combination of the selected anchors does not reduce the objective.

Greedy selection. To solve this optimization problem, we construct the anchor set greedily, starting from $\mathcal{A}_0 = \emptyset$ with $\mathbf{P}_\emptyset = \mathbf{0}$. At iteration t, we select the candidate that gives the largest reduction in the total projection error:

$$a_t \in \operatorname*{arg\,max}_{j\notin\mathcal{A}_{t-1}} \Delta(j \mid \mathcal{A}_{t-1}), \qquad \Delta(j \mid \mathcal{A}) \triangleq \mathcal{E}(\mathcal{A}) - \mathcal{E}(\mathcal{A}\cup\{j\}), \tag{7}$$

and update $\mathcal{A}_t = \mathcal{A}_{t-1}\cup\{a_t\}$. To evaluate this gain efficiently, let $\mathbf{r}_i^{(t-1)} = \hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}_{t-1}}\hat{\mathbf{x}}_i$ denote the residual of token i after projection onto the subspace spanned by the anchors in $\mathcal{A}_{t-1}$. For $\mathbf{r}_j^{(t-1)} \neq \mathbf{0}$, the marginal gain has the following closed form (derived in Appendix A.1):

$$\Delta(j \mid \mathcal{A}_{t-1}) = \sum_{i=1}^{N} w_i \,\frac{\big\langle \mathbf{r}_i^{(t-1)}, \mathbf{r}_j^{(t-1)}\big\rangle^2}{\big\|\mathbf{r}_j^{(t-1)}\big\|_2^2}. \tag{8}$$

Here, j indexes the unselected candidate being evaluated, while i runs over all N original tokens. The gain therefore aggregates the error reduction across the original token set when candidate j is added. It can be evaluated through incremental residual updates without explicitly recomputing the projection matrix. Our reconstruction objective and greedy selection rule instantiate generalized column subset selection (Farahat et al., 2015) with importance-weighted targets.

Segment-wise parallel evaluation. Even with incremental residual updates, evaluating each candidate against all N tokens remains costly for long videos. We therefore partition the visual tokens into G non-overlapping temporal segments $\{\mathcal{S}_1, \ldots, \mathcal{S}_G\}$. We adopt a segment-wise reconstruction objective, projecting each token onto the subspace spanned by anchors in its own segment. We score each candidate by how much its addition reduces the total importance-weighted projection error within that segment.

Formally, before iteration t, let $\mathcal{A}_{g,t-1} = \mathcal{A}_{t-1} \cap \mathcal{S}_g$ be the anchors in segment g, and let $\mathbf{r}_{i,g}^{(t-1)} = \hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}_{g,t-1}} \hat{\mathbf{x}}_i$ be the corresponding local residual for $i \in \mathcal{S}_g$. For a candidate $j \in \mathcal{S}_g \setminus \mathcal{A}_{g,t-1}$ with nonzero residual, its marginal gain under this objective is

$$\Delta_g(j \mid \mathcal{A}_{g,t-1}) = \sum_{i \in \mathcal{S}_g} w_i \frac{\left\langle \mathbf{r}_{i,g}^{(t-1)}, \mathbf{r}_{j,g}^{(t-1)} \right\rangle^2}{\left\| \mathbf{r}_{j,g}^{(t-1)} \right\|_2^2}. \tag{9}$$

Candidate gains are evaluated in parallel across segments, and each iteration selects one anchor globally:

$$(g_t, a_t) \in \underset{\substack{g \in \{1,\ldots,G\} \\ j \in \mathcal{S}_g \setminus \mathcal{A}_{g,t-1}}}{\arg\max} \Delta_g(j \mid \mathcal{A}_{g,t-1}). \tag{10}$$

Only the winning segment updates its anchor subspace and residuals. Selection continues until K anchors have been chosen, allowing the segments to share a single token budget.

3.4 Token Merging

To retain information from unselected tokens without increasing the token budget, we selectively merge them into temporally nearby anchors. For each unselected token, we identify the anchor with the highest cosine similarity within the current frame and the two preceding frames. The token is assigned to this anchor only if their similarity exceeds an adaptive threshold θ, limiting the mixing of dissimilar features. We determine θ for each video from the distribution of each token's maximum cosine similarity to tokens in the preceding frame. Finally, each anchor is updated via an importance-weighted average of itself and its assigned tokens, while unassigned tokens are discarded. Details of the threshold computation and merging procedure are provided in Appendix A.2.

3.5 In-LLM Attention Calibration and Text-Guided Compression

After token merging, the retained tokens may represent different numbers of original visual tokens. To account for these differences, we calibrate the LLM's attention computation using proportional attention (Bolya et al., 2023). Text-to-vision attention also provides a relevance signal for further visual token compression. Following FastV (Chen et al., 2024), we retain visual tokens that receive the highest attention from the last text token at an intermediate LLM layer. The remaining tokens are merged into their most similar retained tokens. Implementation details are provided in Appendix A.3.

4 Experiments

4.1 Experimental Setup

Models and Benchmarks. We implement our method on three representative open-source Video-LLMs: LLaVA-OneVision-7B (Li et al., 2025), LLaVA-Video-7B (Zhang et al., 2025b), and Qwen2.5-VL-7B-Instruct (Bai et al., 2025b). We evaluate on four widely used video understanding benchmarks spanning diverse video lengths and reasoning types: Video-MME (Fu et al., 2025) (without subtitles), LongVideoBench (Wu et al., 2024), MVBench (Li et al., 2024), and EgoSchema (Mangalam et al., 2023). Further details on models and benchmarks are provided in Appendix A.4.

Baselines. We compare against seven recent training-free token compression methods for Video-LLMs that cover the main design paradigms in the literature, including VisionZip (Yang et al., 2025), FastVID (Shen et al., 2025), HoliTom (Shao et al., 2025a), FlashVID (Fan et al., 2026),

UniComp (Yuan et al., 2026), EarlyTom (Wang et al., 2026), and GSTEP (Zhang et al., 2026). We use the official implementations of all baselines, with "–" denoting unsupported model and retention ratio configurations.

Implementation details. All experiments are conducted on NVIDIA A800 or RTX 4090 GPUs. We evaluate all Video-LLMs with the lmms-eval framework (Zhang et al., 2025a). Following the default configurations of each model, we uniformly sample 32 frames per video for LLaVA-OneVision and Qwen2.5-VL, and 64 frames for LLaVA-Video. For LLaVA-Video, we adopt the "one_token" newline setting. Following prior work (Shao et al., 2025b; Fan et al., 2026), we compare all methods at matched layer-average token retention ratios, as defined in Eq. 3. More implementation details are provided in Appendix A.6.

4.2 Main Results

Table 1 compares AVTC with existing token compression methods at layer-average token retention ratios of 2%, 5%, and 10%. AVTC achieves the highest average accuracy on all three backbones at each ratio. Moreover, its average accuracy advantage over the strongest baseline grows as retention decreases from 10% to 2% (Figure 1). Results at higher retention ratios are provided in Appendix A.5.

Results on LLaVA-OneVision-7B. At $\rho = 2\%$, AVTC achieves an average accuracy of 55.26, retaining 94.6% of the uncompressed model's average accuracy and outperforming the strongest baseline by 1.80 points in average accuracy. At 5% retention, it preserves 98.9% of the uncompressed model's average accuracy and outperforms the strongest baseline by 1.23 points in average accuracy. At 10% retention, its average accuracy reaches 58.55, slightly exceeding the uncompressed model's 58.43.

Results on LLaVA-Video-7B and Qwen2.5-VL-7B. AVTC also achieves the highest average accuracy on LLaVA-Video-7B and Qwen2.5-VL-7B-Instruct at all three evaluated retention ratios. As retention decreases from 10% to 2%, the improvement in average accuracy over the strongest baseline increases from 0.91 to 2.39 points on LLaVA-Video-7B and from 0.89 to 2.79 points on Qwen2.5-VL-7B-Instruct. These results support AVTC's effectiveness across the evaluated backbones under aggressive compression.

Inference Efficiency. Table 2 reports inference latency on LLaVA-OneVision-7B at $\rho = 10\%$, measured on a single NVIDIA A800 GPU. At this setting, AVTC achieves 100.2% of the uncompressed model's average accuracy while delivering a $6.6\times$ speedup in total prefill and a $2.0\times$ speedup in time to first token (TTFT). These speedups include compression overhead, and AVTC achieves the lowest total prefill latency and TTFT among the comparable methods. In particular, segment-wise parallel evaluation (SPE) reduces AVTC's compression latency from 113.8 ms to 25.1 ms, a $4.5\times$ speedup over the variant without SPE.

Distributional Analysis. To assess how well compression preserves the feature distribution of the original visual tokens, we compare the compressed and uncompressed token sets on LLaVA-OneVision-7B at 10% retention using Sinkhorn divergence with cosine distance (Feydy et al., 2019) and the squared Wasserstein-2 distance between Gaussian approximations after a fixed random projection (Gelbrich, 1990). These metrics assess discrepancies in the empirical feature distribution and in the mean and covariance of the projected token features, respectively. As shown in Table 3, AVTC achieves the lowest values on both metrics, reducing Sinkhorn divergence and Gaussian W_2^2 by approximately 23.7% and 67.4%, respectively, relative to the best competing score on each metric. These results complement the accuracy comparisons by showing that AVTC also preserves the original visual feature distribution more faithfully under aggressive compression.

4.3 Ablation Studies

We conduct ablation studies on visual-token importance estimation, anchor selection, and token merging on LLaVA-OneVision-7B at $\rho = 5\%$, with results summarized in Table 4. Additional ablation studies are provided in Appendix A.5.

Table 1: **Comparison of state-of-the-art methods on LLaVA-OneVision, LLaVA-Video and Qwen2.5-VL**. Score and Rel. denote the average accuracy across the four benchmarks and performance relative to the uncompressed model, respectively. Best and second-best available results within each model–retention setting are shown in bold and underlined, respectively.

Retention	Method	LLaVA-OneVision-7B						LLaVA-Video-7B						Qwen2.5-VL-7B-Instruct					
		Ego	LVB	MVB	V-MME	Avg.		Ego	LVB	MVB	V-MME	Avg.		Ego	LVB	MVB	V-MME	Avg.	
						Score	Rel.					Score	Rel.					Score	Rel.
100%	Vanilla	60.29	56.55	58.43	58.48	58.43	100.0%	57.64	60.51	61.88	63.48	60.88	100.0%	58.44	58.79	68.62	61.44	61.82	100.0%
10%	VisionZip *(CVPR'25)*	58.20	49.66	53.35	53.67	53.72	91.9%	53.35	53.63	56.78	57.19	55.23	90.7%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	58.56	55.20	57.08	57.37	57.05	97.6%	52.02	55.65	58.63	58.81	56.28	92.4%	55.30	55.35	64.38	55.07	57.52	93.0%
	GSTEP *(ACM MM'26)*	58.26	55.27	56.15	56.85	56.63	96.9%	52.22	55.80	56.55	59.15	55.93	91.9%	53.69	**57.97**	63.18	56.93	57.94	93.7%
	EarlyTom *(CVPR'26)*	60.03	54.75	56.75	56.15	56.92	97.4%	54.18	55.57	58.28	58.67	56.67	93.1%	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	60.03	56.02	57.55	57.30	57.72	98.8%	54.14	57.07	58.05	58.70	56.99	93.6%	55.79	56.55	65.28	57.67	58.82	95.1%
	UniComp *(CVPR'26)*	60.47	**57.14**	56.45	**58.07**	58.03	99.3%	52.06	57.67	56.58	58.44	56.19	92.3%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	61.14	56.02	57.73	57.00	57.97	99.2%	55.34	58.04	58.18	59.81	57.84	95.0%	–	–	–	–	–	–
	AVTC *(Ours)*	**61.58**	56.92	**57.80**	57.89	**58.55**	**100.2%**	**56.33**	**58.86**	**59.08**	**60.74**	**58.75**	**96.5%**	**56.21**	57.52	**66.65**	**58.44**	**59.71**	**96.6%**
5%	VisionZip *(CVPR'25)*	51.82	46.22	45.15	48.41	47.90	82.0%	44.11	48.39	47.53	51.67	47.92	78.7%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	56.95	51.31	53.65	53.74	53.91	92.3%	48.60	53.48	56.35	57.15	53.89	88.5%	52.81	52.95	61.75	52.33	54.96	88.9%
	GSTEP *(ACM MM'26)*	57.21	53.48	55.45	53.85	55.00	94.1%	49.21	55.27	55.38	57.26	54.28	89.2%	52.26	54.75	61.45	54.44	55.72	90.1%
	EarlyTom *(CVPR'26)*	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	59.37	53.18	54.68	54.41	55.41	94.8%	51.56	54.00	55.60	56.26	54.36	89.3%	54.48	54.75	63.60	55.67	57.12	92.4%
	UniComp *(CVPR'26)*	59.05	54.15	54.30	54.30	55.45	94.9%	49.95	56.32	55.00	56.67	54.48	89.5%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	60.90	53.18	**56.98**	55.15	56.55	96.8%	52.91	54.68	56.55	57.33	55.37	90.9%	–	–	–	–	–	–
	AVTC *(Ours)*	**61.98**	**56.17**	56.83	**56.15**	**57.78**	**98.9%**	**55.56**	**57.29**	**57.13**	**58.67**	**57.16**	**93.9%**	**54.70**	**56.62**	**64.85**	**56.22**	**58.10**	**94.0%**
2%	VisionZip *(CVPR'25)*	47.68	45.18	41.68	45.96	45.12	77.2%	37.15	44.43	44.15	47.56	43.32	71.2%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	54.03	47.27	49.28	49.11	49.92	85.4%	42.34	48.99	50.80	52.33	48.62	79.9%	49.63	49.59	55.58	49.04	50.96	82.4%
	GSTEP *(ACM MM'26)*	54.46	51.83	53.63	52.67	53.15	91.0%	46.17	53.48	**54.85**	54.96	52.37	86.0%	50.11	52.21	57.93	51.37	52.90	85.6%
	EarlyTom *(CVPR'26)*	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	54.96	47.72	49.85	49.78	50.58	86.6%	45.50	49.22	52.15	51.33	49.55	81.4%	51.52	50.64	58.88	52.56	53.40	86.4%
	UniComp *(CVPR'26)*	54.96	50.64	51.53	50.04	51.79	88.6%	44.29	50.49	52.73	51.74	49.81	81.8%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	58.10	50.41	53.58	51.74	53.46	91.5%	45.98	52.43	52.98	54.44	51.46	84.5%	–	–	–	–	–	–
	AVTC *(Ours)*	**61.14**	**51.98**	**54.40**	**53.52**	**55.26**	**94.6%**	**54.56**	**54.82**	54.10	**55.56**	**54.76**	**90.0%**	**52.57**	**55.12**	**62.65**	**54.41**	**56.19**	**90.9%**

Table 2: Inference efficiency on LLaVA-OneVision-7B at 10% retention. All values are milliseconds per video.

Method	Vision Encoding	Prefilling Time			TTFT
		Compression	LLM Forward	Total	
Vanilla	382.5	–	578.6	578.6 (1.0×)	961.1 (1.0×)
UniComp	382.1	136.5	66.7	203.2 (2.8×)	585.3 (1.6×)
GSTEP	378.6	133.2	67.8	201.0 (2.9×)	579.6 (1.7×)
FlashVID	378.7	50.9	58.5	109.4 (5.3×)	488.1 (2.0×)
HoliTom	378.5	74.4	97.7	172.1 (3.4×)	550.6 (1.7×)
AVTC	381.9	25.1	62.6	87.7 (6.6×)	469.6 (2.0×)
AVTC(w/o SPE)	380.9	113.8	62.5	176.3 (3.3×)	557.2 (1.7×)

Table 3: Token-set fidelity on LLaVA-OneVision-7B at 10% retention.

Method	Sinkhorn ↓	W_2^2 ↓
VisionZip	0.225	65.46
FastVID	0.200	66.74
EarlyTom	0.137	23.53
FlashVID	0.141	55.46
HoliTom	0.115	12.13
GSTEP	0.108	13.60
UniComp	0.097	14.05
AVTC	**0.074**	**3.96**

Visual-token Importance Estimation. We examine the importance signal used in AVTC's anchor selection and token merging by replacing the diagonal attention score in Eq. (5) with *Incoming Attention*. This alternative averages the off-diagonal attention weights received by each token from other tokens in the same frame, then averages across heads in the final vision-encoder layer. We keep the per-frame min–max normalization and all other settings unchanged. As shown in Table 4, this replacement lowers average accuracy from 57.78 to 57.12, a drop of 0.66 percentage points, with performance decreasing on all four benchmarks. These results support our choice of diagonal attention as an importance proxy for weighting reconstruction errors during set-aware anchor selection and feature contributions during token merging under aggressive compression.

Anchor Token Selection. We next examine anchor selection while keeping the remaining components of AVTC unchanged. Replacing our selection criterion with *Importance Top-k*, which ranks tokens solely by their importance scores, lowers average accuracy by 2.38 points. This drop highlights the limitation of selecting individually important tokens without accounting for information already represented by the selected set. We then compare two alternative greedy selection criteria. The first, *Self Residual*, is inspired by the residual-based selection strategies in ResPrune (Li et al., 2026) and SPARE (Lee et al., 2026). It scores each candidate by the norm of its projection

Table 4: Component ablations on LLaVA-OneVision-7B at 5% retention.

Variant	EgoSchema	LongVideoBench	MVBench	Video-MME	Avg.
Visual-token importance estimation					
Incoming Attention	61.72	54.30	56.40	56.07	57.12
Anchor selection					
Importance Top-k	60.92	52.88	54.10	53.70	55.40
Self Residual	61.90	54.53	56.15	55.63	57.05
Facility Location	62.00	53.70	56.65	55.22	56.89
w/o Importance Weighting	60.70	55.42	55.98	56.48	57.15
w/o SPE	61.76	55.27	56.30	56.37	57.43
Token Merging					
w/o Merging	59.85	55.20	56.48	55.59	56.78
w/o Importance weighting	61.58	55.42	56.33	56.19	57.38
w/o Selective	61.66	56.55	55.90	56.11	57.55
AVTC	61.98	56.17	56.83	56.15	**57.78**

residual relative to the current anchor subspace. This score measures the candidate's own reconstruction error but does not directly assess how much selecting it reduces the reconstruction errors of other tokens. Using this criterion lowers average accuracy by 0.73 points relative to AVTC. The second, *Facility Location*, greedily maximizes the sum of similarities between each token and its best-matching anchor, using the objective adopted by FLoC (Cho et al., 2026). It evaluates coverage through individual anchor matches, whereas AVTC measures the reduction in importance-weighted reconstruction error using linear combinations of the selected anchor features. This variant lowers average accuracy by 0.89 points. We also examine the contribution of importance weighting by setting $w_i = 1$ in AVTC's reconstruction objective (*w/o Importance Weighting*), while retaining importance-weighted token merging. Average accuracy drops by 0.63 points, supporting the benefit of prioritizing reconstruction of important visual features during anchor selection. Finally, we assess segment-wise parallel evaluation (SPE) by replacing it with greedy anchor selection over the entire video ($G = 1$; *w/o SPE*). This change reduces average accuracy from 57.78 to 57.43, a drop of 0.35 points. The result suggests that evaluating reconstruction gains within each segment benefits anchor selection under aggressive compression.

Token Merging. We assess the contribution of token merging by discarding all unselected tokens (*w/o Merging*), which lowers average accuracy by 1.00 point. This result supports incorporating information from unselected tokens into the selected anchors. We then examine the selective merging criterion by removing the similarity threshold (*w/o Selective*). Merging without this filtering lowers average accuracy by 0.23 points, indicating the benefit of restricting aggregation to sufficiently similar tokens. Finally, replacing importance-weighted fusion with uniform averaging of each anchor and its assigned tokens (*w/o Importance Weighting*) reduces average accuracy by 0.40 points, supporting the use of token importance to guide feature aggregation.

5 Conclusion

In this paper, we introduced AVTC, a training-free framework for aggressive visual token compression in Video-LLMs. AVTC greedily selects the token whose addition most reduces the total importance-weighted error of projecting all original visual features onto the subspace spanned by the selected anchors. This criterion favors complementary tokens that jointly represent important visual information under tight token budgets. Experiments across multiple models and benchmarks demonstrate improved accuracy over competing methods under aggressive compression and practical inference speedups. These results support set-aware selection as an effective approach to preserving visual information under aggressive compression.

AI USE STATEMENT

Generative AI tools were used solely for data processing, polishing the writing, and improving the readability of this paper. They were not used for any other tasks, including research ideation, methodology design, implementation, theoretical development, or interpretation of results. All AI-assisted work was reviewed by the authors, and we take full responsibility for the final content of this work.

REFERENCES

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

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

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

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, 2024.

Janghoon Cho, Jungsoo Lee, Munawar Hayat, Kyuwoong Hwang, Fatih Porikli, and Sungha Choi. FLoC: Facility location-based efficient visual token compression for long video understanding. In *International Conference on Learning Representations*, pp. 102973–102996, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a87d3a6b4884108c0dff392c1c7e2d82-Abstract-Conference.html.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. SCOPE: Saliency-coverage oriented token pruning for efficient multimodel LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ec6b4456c2bdfd04002d7984043c4936-Abstract-Conference.html.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. PACT: Pruning and clustering-based token reduction for faster visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14582–14592, 2025.

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*, pp. 98730–98759, 2026. URL https://openreview.net/forum?id=H6rDX4w6Al.

Ahmed K. Farahat, Ahmed Elgohary, Ali Ghodsi, and Mohamed S. Kamel. Greedy column subset selection for large-scale data sets. *Knowledge and Information Systems*, 45(1):1–34, 2015. doi: 10.1007/s10115-014-0801-8.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2681–2690, 2019.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025.

Matthias Gelbrich. On a formula for the L^2 Wasserstein metric between measures on Euclidean and Hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. doi: 10.1002/mana.19901470121.

Jaeyeon Lee, Shunjie Wen, and Dong-Wan Choi. Moving beyond diversity: Visual token pruning as subspace reconstruction for efficient VLMs. *arXiv preprint arXiv:2606.18681*, 2026. URL https://arxiv.org/abs/2606.18681.

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

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. URL https://arxiv.org/abs/2305.06355.

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, 2024.

Xu Li, Yi Zheng, Yuxuan Liang, Zhe Liu, Xiaolei Chen, Haotian Chen, Rui Zhu, and Xiangyang Xue. ResPrune: Text-conditioned subspace reconstruction for visual token pruning in large vision-language models. *arXiv preprint arXiv:2603.21105*, 2026. URL https://arxiv.org/abs/2603.21105.

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.

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

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

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*, volume 38, pp. 135547–135570, 2025a.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. Growing a twig to accelerate large vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20064–20074, 2025b.

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*, volume 38, pp. 123553–123581, 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18992–19001, 2025.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40559–40568, 2026.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28828–28857, 2024.

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.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. UniComp: Rethinking video compression through informational uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18609–18618, 2026.

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.

Mengjie Zhang, Qihui Zhu, Tao Zhang, Shuangwu Chen, Huihuang Qin, Yu Guo, Shenghao Ye, Zijian Wen, Yunpeng Hou, Dong Jin, Xiaobin Tan, Huasen He, and Jian Yang. GSTEP: Global spatio-temporal density-driven visual token pruning for efficient video large language models. *arXiv preprint arXiv:2608.03083*, 2026. URL https://arxiv.org/abs/2608.03083.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025b. URL https://openreview.net/forum?id=EElFGvt39K.

A Appendix

A.1 Marginal Gain Derivation and Segment-wise Selection

Global reconstruction and marginal gain. We derive Eq. (8) directly from the importance-weighted projection error in Eq. (6), using the normalized features $\hat{\mathbf{x}}_i$ defined in Sec. 3.3. Before iteration t, let $\mathcal{U}_{t-1} = \mathrm{span}\{\hat{\mathbf{x}}_k : k \in \mathcal{A}_{t-1}\}$. For a candidate $j \notin \mathcal{A}_{t-1}$ with $\mathbf{r}_j^{(t-1)} \neq \mathbf{0}$, define $\mathbf{u}_j^{(t-1)} = \mathbf{r}_j^{(t-1)} / \|\mathbf{r}_j^{(t-1)}\|_2$. Since $\mathbf{r}_j^{(t-1)}$ is orthogonal to $\mathcal{U}_{t-1}$ and $\hat{\mathbf{x}}_j = \mathbf{P}_{\mathcal{A}_{t-1}}\hat{\mathbf{x}}_j + \mathbf{r}_j^{(t-1)}$, adding j expands the anchor subspace by exactly the direction $\mathbf{u}_j^{(t-1)}$. The updated orthogonal projector is therefore

$$\mathbf{P}_{\mathcal{A}_{t-1}\cup\{j\}} = \mathbf{P}_{\mathcal{A}_{t-1}} + \mathbf{u}_j^{(t-1)}\big(\mathbf{u}_j^{(t-1)}\big)^\top. \tag{11}$$

Using $\langle \hat{\mathbf{x}}_i, \mathbf{u}_j^{(t-1)} \rangle = \langle \mathbf{r}_i^{(t-1)}, \mathbf{u}_j^{(t-1)} \rangle$, the residual of token i after adding j becomes

$$\widetilde{\mathbf{r}}_i \triangleq \hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}_{t-1}\cup\{j\}}\hat{\mathbf{x}}_i = \mathbf{r}_i^{(t-1)} - \big\langle \mathbf{r}_i^{(t-1)}, \mathbf{u}_j^{(t-1)} \big\rangle \mathbf{u}_j^{(t-1)}. \tag{12}$$

Since $\widetilde{\mathbf{r}}_i$ is orthogonal to the unit vector $\mathbf{u}_j^{(t-1)}$, the Pythagorean theorem yields

$$\begin{aligned}
\Delta(j \mid \mathcal{A}_{t-1}) &= \sum_{i=1}^{N} w_i \left(\big\|\mathbf{r}_i^{(t-1)}\big\|_2^2 - \big\|\widetilde{\mathbf{r}}_i\big\|_2^2 \right) \\
&= \sum_{i=1}^{N} w_i \big\langle \mathbf{r}_i^{(t-1)}, \mathbf{u}_j^{(t-1)} \big\rangle^2 \\
&= \sum_{i=1}^{N} w_i \, \frac{\big\langle \mathbf{r}_i^{(t-1)}, \mathbf{r}_j^{(t-1)} \big\rangle^2}{\big\|\mathbf{r}_j^{(t-1)}\big\|_2^2},
\end{aligned} \tag{13}$$

which recovers Eq. (8). If $\mathbf{r}_j^{(t-1)} = \mathbf{0}$, then $\hat{\mathbf{x}}_j \in \mathcal{U}_{t-1}$, so adding j leaves the subspace unchanged and $\Delta(j \mid \mathcal{A}_{t-1}) = 0$. Thus, the marginal gain measures the total importance-weighted residual energy captured by the newly added direction across all tokens.

Segment-wise reconstruction and selection. The segment-wise procedure in Sec. 3.3 restricts each token's reconstruction to anchors in its own temporal segment. For an anchor set $\mathcal{A}$, define

$$\mathcal{E}_{\mathrm{seg}}(\mathcal{A}) = \sum_{g=1}^{G} \sum_{i \in \mathcal{S}_g} w_i \left\| \hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}\cap\mathcal{S}_g}\hat{\mathbf{x}}_i \right\|_2^2. \tag{14}$$

The selection budget is shared across segments through $|\mathcal{A}| = K$. For $G = 1$, Eq. 14 reduces to Eq. 6. With multiple segments, it restricts the available reconstruction subspaces and generally yields different candidate gains. For $j \in \mathcal{S}_g \setminus \mathcal{A}_{t-1}$, adding j changes only the error in segment g, so

$$\mathcal{E}_{\mathrm{seg}}(\mathcal{A}_{t-1}) - \mathcal{E}_{\mathrm{seg}}(\mathcal{A}_{t-1} \cup \{j\}) = \Delta_g(j \mid \mathcal{A}_{g,t-1}). \tag{15}$$

Applying the preceding derivation to the tokens and anchors in $\mathcal{S}_g$ gives Eq. 9.

Algorithm 1 summarizes this procedure. For each segment $\mathcal{S}_g$, it maintains the current anchor set $\mathcal{A}_g$ and local residuals $\mathbf{r}_i = \hat{\mathbf{x}}_i - \mathbf{P}_{\mathcal{A}_g}\hat{\mathbf{x}}_i$ for $i \in \mathcal{S}_g$. Only the winning segment needs a residual update.

A.2 Details of Selective Token Merging

Adaptive similarity threshold. For each token after the first frame, we compute its highest cosine similarity to the tokens in the preceding frame. We set θ to the 10th percentile of these values across the video, clipped to a fixed range:

$$\theta = \mathrm{clip}\left(Q_{0.1}\left(\left\{ \max_{k \in \mathcal{V}_{t-1}} \langle \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_k \rangle \;\middle|\; i \in \mathcal{V}_t,\ t > 1 \right\} \right), 0.1,\ 0.7 \right), \tag{16}$$

where $Q_{0.1}$ denotes the 10th percentile, $\mathcal{V}_t$ is the set of token indices in frame t, and $\hat{\mathbf{x}}_i$, $\hat{\mathbf{x}}_k$ are the normalized features defined in Sec. 3.3, so their inner product equals cosine similarity. This adapts the merging criterion to each video's temporal feature consistency.

Algorithm 1 Segment-Parallel Greedy Anchor Selection

Require: ℓ_2-normalized features $\{\hat{\mathbf{x}}_i\}_{i=1}^N$, importance weights $\{w_i\}_{i=1}^N$ with $w_i \geq 0$, disjoint segments $\{\mathcal{S}_g\}_{g=1}^G$ partitioning $\{1, \ldots, N\}$, budget $0 \leq K \leq N$
Ensure: Anchor set $\mathcal{A}$ with $|\mathcal{A}| = K$
$\mathcal{A}_g \leftarrow \emptyset$ for all g; $\mathbf{r}_i \leftarrow \hat{\mathbf{x}}_i$ for all i
for $t = 1$ **to** K **do**
  for each g with $\mathcal{S}_g \setminus \mathcal{A}_g \neq \emptyset$ **in parallel do**
    $\mathcal{C}_g \leftarrow \mathcal{S}_g \setminus \mathcal{A}_g$
    Compute $\Delta_g(j \mid \mathcal{A}_g)$ for all $j \in \mathcal{C}_g$ using Eq. 9, assigning zero gain when $\mathbf{r}_j = \mathbf{0}$
    $j_g^\star \leftarrow \arg\max_{j \in \mathcal{C}_g} \Delta_g(j \mid \mathcal{A}_g)$
  end for
  $g^\star \leftarrow \arg\max_{g:\, \mathcal{S}_g \setminus \mathcal{A}_g \neq \emptyset} \Delta_g(j_g^\star \mid \mathcal{A}_g)$; $a_t \leftarrow j_{g^\star}^\star$
  if $\|\mathbf{r}_{a_t}\|_2 > 0$ **then**
    $\mathbf{u} \leftarrow \mathbf{r}_{a_t} / \|\mathbf{r}_{a_t}\|_2$
    $\mathbf{r}_i \leftarrow \mathbf{r}_i - \langle \mathbf{r}_i, \mathbf{u} \rangle \mathbf{u}$ for all $i \in \mathcal{S}_{g^\star}$
  end if
  $\mathcal{A}_{g^\star} \leftarrow \mathcal{A}_{g^\star} \cup \{a_t\}$
end for
return $\mathcal{A} = \bigcup_{g=1}^G \mathcal{A}_g$

Importance-weighted fusion. Let $\mathcal{C}_a$ be the set of unselected tokens assigned to anchor a, and w_i the importance score of token i from Sec. 3.2. The merged token is computed from the original, unnormalized features:

$$\tilde{\mathbf{x}}_a = \frac{w_a \mathbf{x}_a + \sum_{j \in \mathcal{C}_a} w_j \mathbf{x}_j}{w_a + \sum_{j \in \mathcal{C}_a} w_j}. \tag{17}$$

The merged anchor represents $s_a = 1 + |\mathcal{C}_a|$ original visual tokens. We use this count for attention calibration within the LLM.

A.3 Details of the In-LLM Stage

Attention calibration. After vision-side compression, retained anchors may represent different numbers of original visual tokens. Ignoring these differences can underrepresent anchors that aggregate multiple original tokens in the attention distribution. Following the proportional attention mechanism in ToMe (Bolya et al., 2023), we track a size counter s_i for each token i, denoting the number of original visual tokens it represents. The self-attention in every LLM layer is then calibrated as:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_{\mathrm{head}}}} + \alpha \log \mathbf{s} \right) V, \tag{18}$$

where $\mathbf{s} \in \mathbb{R}^n$ is the row vector of token sizes broadcast over queries, d_{head} is the attention head dimension, and α controls the strength of attention calibration. When $\alpha = 1$, this is equivalent to including s_j identical copies of that key in the softmax, while $\alpha = 0$ recovers standard attention.

Text-Guided Secondary Compression. The vision-side compression is text-agnostic. Anchor tokens are selected independently of the user query. Consequently, the retained token set may still contain content irrelevant to the specific question. We therefore perform secondary text-conditioned compression inside the LLM, where text-to-vision attention becomes available. Let ℓ_{c} denote the compression layer and $\rho_{\mathrm{c}} \in (0, 1]$ the fraction of visual tokens retained at this stage. Following FastV (Chen et al., 2024), at layer ℓ_{c}, we compute the importance of each visual token v_j as the attention it receives from the last text token, averaged over all heads:

$$\phi(v_j) = \frac{1}{H} \sum_{h=1}^{H} A_{t_{\mathrm{last}}, j}^{(h)}, \tag{19}$$

where t_{last} is the index of the last text token, H is the number of attention heads, and $A_{t_{\mathrm{last}}, j}^{(h)}$ is the post-softmax attention weight from this text token to visual token v_j at head h of layer ℓ_{c}. We

Table 5: Results at 25%, 20%, and 15% retention across three models. Best and second-best available results within each model are shown in bold and underlined, respectively. Rel. denotes performance relative to the full-token vanilla model.

Retention	Method	LLaVA-OneVision-7B						LLaVA-Video-7B						Qwen2.5-VL-7B-Instruct					
		Ego	LVB	MVB	V-MME	Avg.		Ego	LVB	MVB	V-MME	Avg.		Ego	LVB	MVB	V-MME	Avg.	
						Score	Rel.					Score	Rel.					Score	Rel.
25%	VisionZip *(CVPR'25)*	60.37	56.92	57.73	58.56	58.39	99.9%	57.09	58.56	60.90	60.74	59.32	97.4%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	59.49	56.40	58.05	57.59	57.88	99.1%	55.22	56.99	60.58	61.93	58.68	96.4%	56.89	57.44	66.08	57.89	59.57	96.4%
	GSTEP *(ACM MM'26)*	59.37	57.89	56.78	58.85	58.22	99.6%	55.48	59.09	58.80	61.56	58.73	96.5%	55.85	**58.86**	65.25	59.52	59.87	96.8%
	EarlyTom *(CVPR'26)*	61.02	56.40	57.60	58.67	58.42	100.0%	57.05	59.09	60.23	61.56	59.48	97.7%	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	60.41	56.55	57.93	58.85	58.43	100.0%	57.11	59.61	60.75	61.52	59.75	98.1%	57.09	57.97	67.60	60.07	60.68	98.2%
	UniComp *(CVPR'26)*	61.42	**58.19**	57.80	59.00	59.10	101.1%	52.69	58.94	58.98	60.89	57.87	95.1%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	61.66	55.95	**58.40**	58.59	58.65	100.4%	**57.98**	58.86	60.03	62.00	59.72	98.1%	–	–	–	–	–	–
	AVTC *(Ours)*	**62.29**	57.82	57.65	**59.52**	**59.32**	**101.5%**	57.94	**59.69**	**60.98**	**62.07**	**60.17**	**98.8%**	**57.50**	58.79	**67.83**	**60.30**	**61.10**	**98.8%**
20%	VisionZip *(CVPR'25)*	59.93	55.42	57.63	57.63	57.65	98.7%	56.23	58.04	60.15	60.63	58.76	96.5%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	59.37	57.22	58.05	57.37	58.00	99.3%	54.32	56.92	60.40	61.22	58.22	95.6%	56.37	57.22	66.05	57.07	59.18	95.7%
	GSTEP *(ACM MM'26)*	59.13	57.22	56.68	58.67	57.92	99.1%	54.46	58.71	58.35	60.41	57.98	95.2%	55.32	58.41	64.73	59.52	59.49	96.2%
	EarlyTom *(CVPR'26)*	60.54	55.42	57.55	58.37	57.97	99.2%	56.83	59.16	59.70	60.67	59.09	97.1%	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	60.09	**57.44**	58.10	57.81	58.36	99.9%	56.47	59.46	60.15	60.81	59.22	97.3%	57.05	58.12	67.43	**59.70**	60.57	98.0%
	UniComp *(CVPR'26)*	60.96	57.29	57.30	**58.89**	58.61	100.3%	52.91	59.16	57.93	60.44	57.61	94.6%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	61.02	57.07	**58.43**	58.85	**58.84**	**100.7%**	57.40	58.64	59.60	61.15	59.20	97.2%	–	–	–	–	–	–
	AVTC *(Ours)*	**61.12**	57.22	57.58	58.19	58.52	100.2%	**57.82**	**59.69**	**60.70**	**61.30**	**59.88**	**98.4%**	**57.38**	**58.86**	**67.50**	59.44	**60.80**	**98.3%**
15%	VisionZip *(CVPR'25)*	60.01	54.53	56.60	56.07	56.80	97.2%	55.16	56.02	58.50	60.22	57.48	94.4%	–	–	–	–	–	–
	FastVID *(NeurIPS'25)*	58.86	56.47	57.98	57.70	57.75	98.8%	53.65	56.92	59.60	60.56	57.68	94.7%	56.13	56.40	65.58	56.48	58.65	94.9%
	GSTEP *(ACM MM'26)*	58.99	57.37	56.63	57.74	57.68	98.7%	53.43	58.04	57.68	59.44	57.15	93.9%	54.62	58.19	64.38	58.22	58.85	95.2%
	EarlyTom *(CVPR'26)*	60.50	54.53	57.80	57.81	57.66	98.7%	55.56	56.62	59.18	60.22	57.89	95.1%	–	–	–	–	–	–
	FlashVID *(ICLR'26)*	60.25	**57.67**	57.45	58.22	58.40	99.9%	55.20	58.41	59.48	60.30	58.35	95.8%	56.47	56.99	**66.98**	58.74	59.79	96.7%
	UniComp *(CVPR'26)*	**60.82**	57.37	56.38	58.19	58.19	99.6%	52.53	57.44	56.48	59.93	56.59	93.0%	–	–	–	–	–	–
	HoliTom *(NeurIPS'25)*	**60.82**	55.35	**58.58**	57.81	58.14	99.5%	56.19	58.12	59.15	**61.07**	58.63	96.3%	–	–	–	–	–	–
	AVTC *(Ours)*	60.64	57.07	57.75	**58.81**	**58.57**	**100.2%**	**57.11**	**59.24**	**59.80**	60.78	**59.23**	**97.3%**	**57.03**	**58.27**	66.70	**58.93**	**60.23**	**97.4%**

retain a fraction ρ_c of the visual tokens present at this layer, selecting those with the highest $\phi(\cdot)$, and merge each remaining token into the retained token with the highest cosine similarity.

A.4 Details of Models and Benchmarks

Models. LLaVA-OneVision-7B (Li et al., 2025) encodes each frame with SigLIP into 196 tokens after bilinear pooling and uses Qwen2-7B as the language backbone. LLaVA-Video-7B (Zhang et al., 2025b) shares the same architecture but is further trained on large-scale synthetic video instruction data and samples more frames per video. Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) adopts a native-resolution vision encoder with 3D patch merging, so that the number of tokens per frame varies with the input resolution. Covering both fixed- and dynamic-length visual tokenization allows us to verify that our method is agnostic to the underlying tokenization scheme. All models are used with their official checkpoints and default inference settings.

Benchmarks. Video-MME (Fu et al., 2025) contains 900 videos evenly divided into short (<2 min), medium (4–15 min), and long (30–60 min) subsets with 2,700 multiple-choice questions. We report results without subtitles. LongVideoBench (Wu et al., 2024) targets long-form understanding with videos up to one hour and questions that require referring to specific moments, we evaluate on the validation set. MVBench (Li et al., 2024) consists of 20 temporal reasoning tasks on short clips, emphasizing fine-grained motion and event understanding. EgoSchema (Mangalam et al., 2023) comprises three-minute egocentric videos with questions that demand long-range temporal reasoning, and we report results on the full test set through the official evaluation server. All benchmarks are evaluated as multiple-choice question answering.

A.5 More experimental results

Results at High Token Retention Ratios. Table 5 reports results at higher token retention ratios of 15%, 20%, and 25%. AVTC achieves the highest average accuracy in eight of the nine model–retention settings. The exception is LLaVA-OneVision-7B at 20% retention, where AVTC still slightly exceeds the uncompressed model's average accuracy. These results demonstrate that AVTC remains effective beyond the aggressive compression settings considered in the main text.

Table 6: Inference efficiency and accuracy on LLaVA-Video-7B at 10% retention.

Method	Vision Encoding	Prefilling Time			TTFT	Avg.	
		Compression	LLM Forward	Total		Score	Rel.
Vanilla	643.1	–	1046.2	1046.2 (1.0×)	1689.3 (1.0×)	60.88	100.0%
UniComp	640.8	229.6	98.4	328.0 (3.2×)	968.8 (1.7×)	56.19	92.3%
GSTEP	635.8	208.1	97.9	306.0 (3.4×)	941.8 (1.8×)	55.93	91.9%
FlashVID	636.7	80.0	98.9	178.9 (5.8×)	815.6 (2.1×)	56.99	93.6%
HoliTom	635.1	150.4	160.1	310.5 (3.4×)	945.6 (1.8×)	57.84	95.0%
AVTC	642.8	38.0	96.5	134.4 (7.8×)	777.2 (2.2×)	58.75	96.5%

Table 7: LLM-side component ablations on LLaVA-OneVision-7B at 5% retention.

Variant	EgoSchema	LongVideoBench	MVBench	Video-MME	Avg.
LLM-side refinement					
w/o Attention Calibration	60.17	54.82	55.98	56.59	56.89
w/o Text-guided Compression	61.62	54.97	56.05	56.78	57.36
AVTC	61.98	56.17	56.83	56.15	57.78

Inference Efficiency on LLaVA-Video-7B. Table 6 reports inference efficiency on LLaVA-Video-7B at 10% retention, together with the corresponding average accuracy and relative performance from Table 1. AVTC preserves 96.5% of the uncompressed model's average accuracy while achieving a $7.8\times$ prefill speedup and a $2.2\times$ speedup in TTFT.

Ablation on LLM-side Components. We further ablate attention calibration and text-guided compression to examine how they complement vision-side compression. First, removing attention calibration (*w/o Attention Calibration*) lowers average accuracy by 0.89 points (Table 7). Without this correction, anchors that aggregate more original tokens can be underrepresented in the attention distribution. Calibration accounts for token sizes to mitigate this imbalance. In addition to correcting attention after merging, the LLM-side stage introduces textual guidance into token retention. Removing text-guided compression (*w/o Text-guided Compression*) reduces average accuracy from 57.78 to 57.36. This suggests that using text-to-vision attention to prioritize content relevant to the input question complements the text-independent selection performed on the vision side.

Qualitative Analysis of Anchor Selection. To complement the quantitative ablations in Sec. 4.3, Figure 3 compares *Importance Top-k*, *Self Residual*, *Facility Location*, and AVTC on an MVBench material-recognition example using LLaVA-OneVision-7B at 5% retention. The question asks for the material of the cylinder that is stationary at the beginning of the video. We visualize selected anchors across six frames, with the queried cylinder highlighted in the first four.

AVTC retains patches on and near the queried cylinder. In the second displayed frame, an AVTC anchor overlaps a bright highlight on the cylinder's surface, which is not covered by the anchors selected by the three alternative criteria. AVTC correctly predicts metal, whereas the other variants predict rubber. This example illustrates how set-aware selection can retain visual details relevant to material recognition.

A.6 Reproduction details

Hyperparameter Setting. For segment-wise parallel evaluation, we set the number of frames per segment to 8, which yields $G = 4$ segments for LLaVA-OneVision-7B and Qwen2.5-VL-7B-Instruct, and $G = 8$ segments for LLaVA-Video-7B. The strength α of the attention calibration within the LLM is set to 0.625. Following the budget configuration of FlashVID (Fan et al., 2026), we set the input token budget to $K = 1.25\rho N$ and retain a fraction $\rho_c = 0.3$ of visual tokens after layer $\ell_c = 20$. This configuration matches the target layer-average retention ratio ρ.

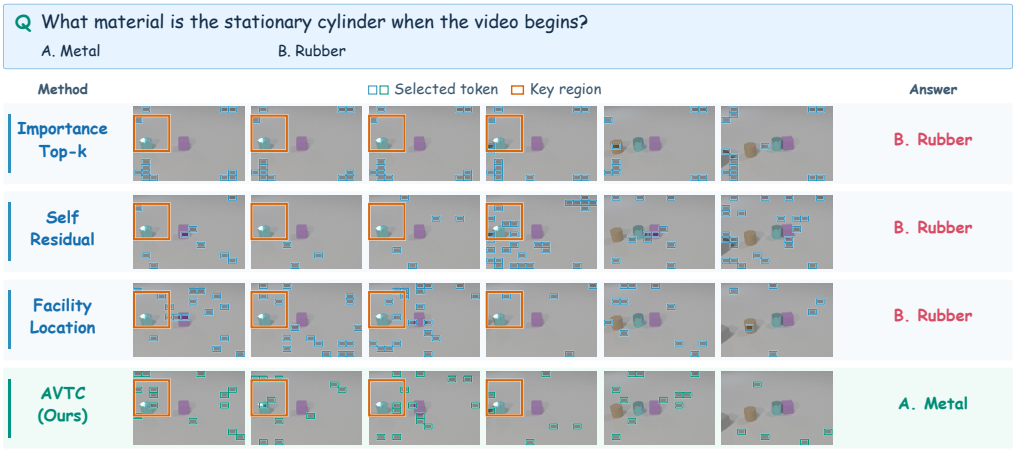

Figure 3: **Anchor selection for material recognition at** 5% **retention.** Each row shows the anchors selected by one method on the same six video frames, with the corresponding model prediction on the right. Blue and green outlines mark anchors selected by the baselines and AVTC, respectively; orange rectangles highlight the queried cylinder in the first four frames. In the second displayed frame, AVTC retains a patch overlapping the cylinder's surface highlight. AVTC correctly predicts metal, while the other variants predict rubber.