FOVEATREE: INFORMATION-DENSITY TREES FOR QUERY-GUIDED VISUAL TOKEN PRUNING IN VIDEO LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video large language models (Video LLMs) have demonstrated strong multimodal understanding across a range of tasks, including visual question answering and video captioning. Yet, encoding each frame into hundreds of visual tokens makes inference over temporal visual content expensive. Existing methods reduce this cost by merging similar tokens or using text-to-vision attention to select task-relevant tokens. Nevertheless, these methods still face two main challenges: (1) **Hierarchy-blind Pruning**: many similarity-based methods rely on pairwise similarity or training a lightweight neural network to predict importance scores without explicitly considering the regional compressibility of a frame, which can misguide the merging or removal of critical evidence. (2) **Attention Drift**: position-dependent effects from Rotary Position Embedding (RoPE), which encodes token positions in the decoder, together with preferences for nearby or recent tokens and attention sinks can shift query-to-vision attention away from task relevance, causing attention-based pruning to underweight and discard task-relevant visual tokens. Motivated by these observations, we propose FoveaTree, a two-stage, training-free visual-token pruning framework for efficient video inference. The first stage estimates regional compressibility from the feature distortion incurred when merging a region to construct a quadtree-derived density map, which guides the dropping, merging, and retention of visual regions. Based on the surviving units and inherited tree topology, the second stage introduces an efficient tree-structured pruning mechanism in the middle layers of the LLM to mitigate Attention Drift and further reduce low-relevance visual content using text-to-vision attention. We validate FoveaTree under different Video LLM architectures, i.e., InternVL3.5-8B and Qwen3-VL-8B, on five video question-answering benchmarks and one captioning benchmark with token budgets of 50%, 30%, and 10%; our method consistently achieves overall state-of-the-art performance across all evaluated settings, demonstrating a superior accuracy-efficiency trade-off.[1]

1 INTRODUCTION

Recent video large language models (Video LLMs) (Wang et al., 2025; Bai et al., 2025a) exhibit strong video understanding, delivering competitive performance on tasks such as video question answering and video captioning (Li et al., 2025; Bai et al., 2025b). A Video LLM contains a vision encoder, a modality projector, and an LLM (Liu et al., 2023; Lin et al., 2024; Li et al., 2024b); the encoder represents each frame as patch tokens, which the projector maps into the language embedding space (Dosovitskiy et al., 2021; Radford et al., 2021; Zhai et al., 2023; Oquab et al., 2024; He et al., 2022; Liu et al., 2021). Since the visual sequence grows with video length and resolution (Wang et al., 2024a; Xu et al., 2024), it inflates attention computation, prefill latency, and key-value cache memory, motivating visual-token pruning for practical inference (Tu et al., 2025).

Visual-token pruning lowers this cost (Shang et al., 2025; Endo et al., 2025; Zou et al., 2025; Xing et al., 2025; Arif et al., 2025; Zhang et al., 2025a). ToMe (Bolya et al., 2023) and EViT (Liang et al., 2022) merge or discard redundant tokens (Ryoo et al., 2021; Rao et al., 2021; Fayyaz et al., 2022;

[1] Code will be released in the final version of the paper.

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.

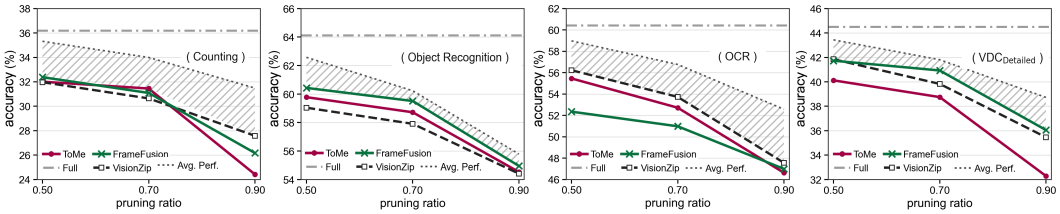


Figure 1: Illustration of Hierarchy-blind Pruning on fine-grained domains from VideoMME (Fu et al., 2025a) and VDC (Chai et al., 2025). Avg. Perf. summarizes performance retention on the complete evaluation set and is projected onto each domain's full-model score, whereas the method curves report the measured domain-level performance. The gray regions show the resulting gap on OCR, counting, object recognition, and detailed captioning across pruning ratios.

Marin et al., 2023); VisionZip (Yang et al., 2025), FrameFusion (Fu et al., 2025b), and STTM (Hyun et al., 2025) extend merging to cross-frame and spatiotemporal redundancy; and FastV (Chen et al., 2024a), MADTP (Cao et al., 2024), and LearnPruner (Takezoe et al., 2026) select query-related tokens via text-to-vision attention (Ye et al., 2025; Zhang et al., 2025b). These methods shorten visual sequences but provide limited control over how evidence is organized (Wen et al., 2025).

Fig. 1 compares overall and domain-level retention for ToMe, VisionZip, and FrameFusion. Avg. Perf. accounts for general performance degradation at each budget, whereas the method curves report measured domain performance. The gray regions therefore isolate additional losses of fine-grained evidence rather than a uniform decline of the model. The panels test small text (OCR), distinct instances (counting), local appearance and boundaries (object recognition), and multiple visual details within one description (VDC detailed captioning). All require coherent evidence across spatial scales, yet similarity methods make pairwise token-level decisions without representing how cells form nested regions. Their systematic domain deficit thus exposes a blind spot to regional hierarchy, which we term **Hierarchy-blind Pruning**. Fig. 2 illustrates this limitation with FrameFusion, a representative similarity-based method. FrameFusion computes cosine similarity only between spa-

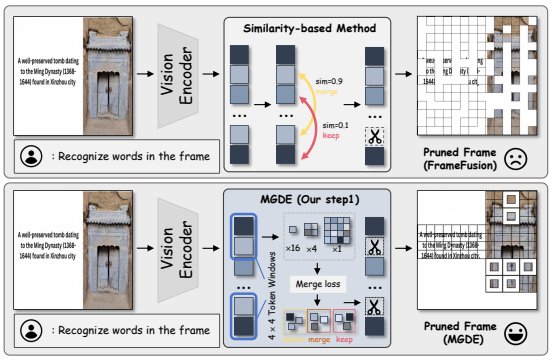


Figure 2: Architectural comparison of FrameFusion and MGDE. FrameFusion merges corresponding tokens across adjacent frames; MGDE instead constructs an explicit multiscale regional hierarchy.

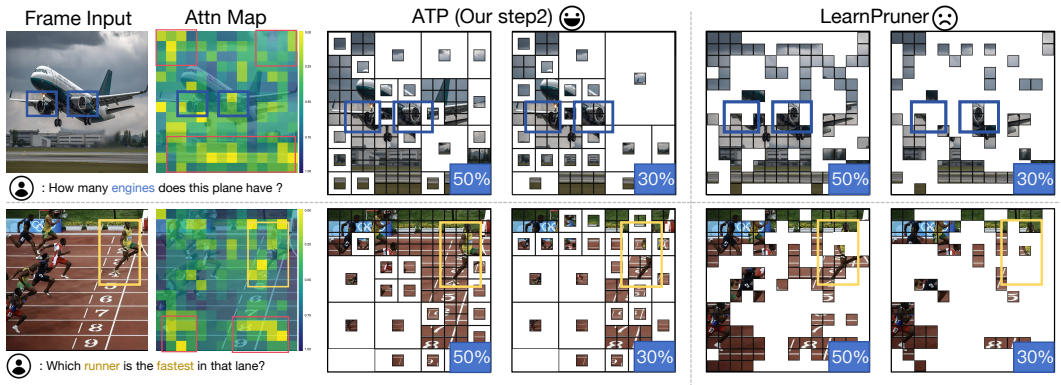


Figure 3: Visualization of our tree-based attention pruning and the conventional attention-based top-k pruning used by LearnPruner (Takezoe et al., 2026) at 50% and 30% visual-token retention for two input frames. The comparison illustrates Attention Drift around peripheral and task-irrelevant regions (red boxes), while the tree structure improves robustness to this bias at both budgets.

tially corresponding tokens in adjacent frames and performs cascaded merging, but it neither estimates within-frame regional compressibility nor constructs a nested spatial hierarchy.

Fig. 3 visualizes query-to-vision attention scores. **Attention Drift** appears as anomalously high responses at spatial boundaries or early visual-sequence positions despite weak query relevance, reflecting RoPE positional effects, nearby or recent-token preferences, and attention sinks (Su et al., 2024; Xiao et al., 2024; Zhao et al., 2025a; Zhang et al., 2026; Liu et al., 2024b). Flat top-k methods such as LearnPruner follow these peaks; as retention falls from 50% to 30%, query-relevant regions become increasingly fragmented or lost, whereas ATP uses the inherited tree to preserve relevant regions together with their context, keeping the surrounding visual evidence intact during pruning.

Motivated by these observations, we propose FoveaTree, a two-stage, training-free visual-token pruning framework. Merge-Gain Density Estimation (MGDE) builds a quadtree for each frame, estimates compressibility from merge-induced feature distortion, labels terminal regions by density, and performs drop, merge, and keep operations. Attention Tree Pruning (ATP) then uses middle-layer text-to-vision attention to prune the surviving hierarchy to the target budget per frame.

Together, the two stages preserve core regions more precisely, address the challenges above, and maintain spatial context during query-guided pruning. We validate FoveaTree under different Video LLM architectures, InternVL3.5-8B and Qwen3-VL-8B, across five video question-answering benchmarks and one captioning benchmark at 50%, 30%, and 10% visual-token retention. Across all evaluated settings, FoveaTree achieves overall state-of-the-art performance and demonstrates a superior accuracy-efficiency trade-off across both backbones and all retention budgets.

In summary, our main contributions are as follows:

- We propose FoveaTree, a two-stage, training-free visual-token pruning framework for efficient Video LLM inference. By combining compressibility-guided spatial pruning with query-guided tree pruning within the LLM, FoveaTree efficiently preserves task-relevant visual evidence at low visual-token retention without additional training or labels.
- We introduce MGDE, which builds a multiscale hierarchy and density map from merge distortion and gain, then adapts pruning to regional compressibility by dropping, merging, or retaining tokens. We also introduce ATP to propagate attention scores through the inherited quadtree, improving robustness to Attention Drift and preserving task-relevant regions.
- We conduct experiments on five video question-answering benchmarks and one captioning benchmark to evaluate FoveaTree across different Video LLM architectures, while component ablations on InternVL3.5-8B assess each proposed module. Notably, at 30% visual-token retention on InternVL3.5-8B, FoveaTree achieves 100.62% performance of the full model, with a $1.24\times$ inference speedup and 53.2% lower TFLOPs.

2 METHOD

2.1 PRELIMINARY

Let M_θ denote a Video LLM with a vision encoder, multimodal projector, L-layer language decoder, and output projection. Given a video $v = \{v_t\}_{t=1}^{T}$ and query q, the encoder and projector map each frame to an LLM-aligned grid $X_t = \{x_{t,p} \mid p \in \mathcal{P}\}$, where $\mathcal{P} = \{1, \dots, H\} \times \{1, \dots, W\}$ and $x_{t,p} \in \mathbb{R}^d$. The visual tokens are concatenated with the embedded query, and the decoder predicts $p_\theta(\cdot \mid v, q, y_{<r}) = \mathrm{softmax}(f_\theta(\cdot \mid v, q, y_{<r}))$ before sampling the next token y_r. The original visual prefix contains $N = T|\mathcal{P}| = THW$ tokens, so prefill computation and KV-cache memory grow with N. Given a retention ratio $0 < \eta \leq 1$, we define the nominal per-frame budget as $B = \lceil \eta HW \rceil$ from the original visual grid. We denote projected features by $x_{t,p}$ and shallow vision features by $z_{t,p}$; MGDE uses the latter for spatial estimation before decoder prefill in every frame.

2.2 FOVEATREE FRAMEWORK

As shown in Fig. 4, FoveaTree reduces visual tokens in two stages: regional compression before decoder prefill and query-guided pruning within the decoder. MGDE first uses layer-0 vision features to construct a quadtree forest $\mathcal{T}_t$ per frame. The sizes of its terminal regions then define an operational density map $\mathcal{M}_t$, which guides dropping, merging, and keeping of projected visual tokens,

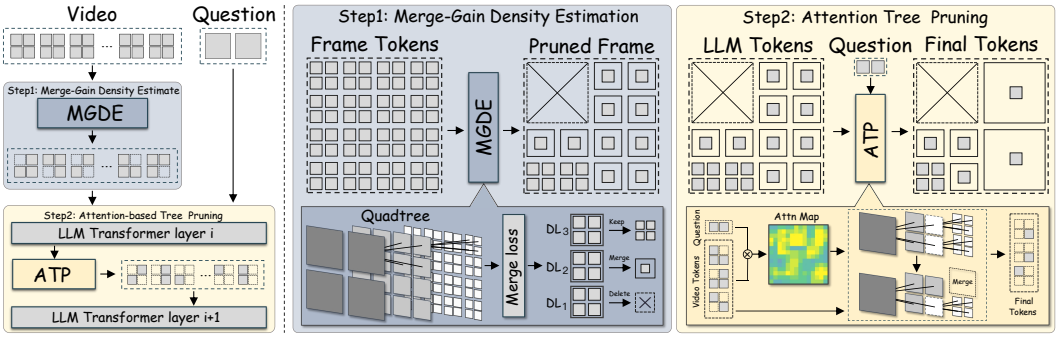


Figure 4: FoveaTree overview. MGDE builds a regional hierarchy and compresses visual tokens before decoder prefill; ATP then uses text-to-vision attention to prune the surviving hierarchy.

producing a compact unit set $\mathcal{V}$ whose elements retain their spatial support and ancestry in $\mathcal{T}_t$. At an intermediate decoder layer, ATP assigns text-to-vision attention scores to $\mathcal{V}$ and propagates them over the surviving branches of $\mathcal{T}_t$. A top-down cut selects the resolution of each region under the per-frame budget during decoder processing, linking prefill compression with query-guided pruning.

2.3 Merge-Gain Density Estimation

Motivation. As discussed above, similarity-based pruning can merge or discard tokens from informative regions because it does not explicitly represent the spatial hierarchy of a frame, and learned selectors require training and repeated model evaluation. MGDE provides a training-free alternative by measuring the feature distortion caused by regional merging. It uses this signal to construct a multiscale partition, then derives a density map that assigns finer or coarser token resolution according to the terminal region size before the visual tokens enter the language decoder for prefill.

Regional compressibility. MGDE computes regional compressibility from shallow features $z_{t,p}$. Let $\bar{z}_{t,p} = z_{t,p}/(\|z_{t,p}\|_2 + \epsilon)$, where $\epsilon > 0$ is a numerical constant. For a quadtree region R, let $\mathcal{B}(R) = R \cap \mathcal{P}$ be its valid grid cells. We define the distortion of representing R with one token as $\mathcal{D}_t(R) = |\mathcal{B}(R)| - \left\| \sum_{p \in \mathcal{B}(R)} \bar{z}_{t,p} \right\|_2$. If $R^{(1)}, \ldots, R^{(4)}$ are its four quadrants, the gain from preserving their separate representations is $G_t(R) = \mathcal{D}_t(R) - \sum_{k=1}^{4} \mathcal{D}_t\big(R^{(k)}\big)$. Low distortion indicates that a region is suitable for coarse representation, whereas a large merge gain indicates greater feature variation within the region and local structure that should remain separated. To determine which vision features best support MGDE, we extract features from multiple encoder layers and use each of them for regional analysis. Fig. 5 shows a consistent pattern across three vision encoders: layer-0 features most clearly resolve frame complexity and its information-density hierarchy across the high-, medium-, and low-density regions, because they preserve local textures and boundaries whereas deeper features become more semantic and less spatially localized. Based on this pattern, MGDE uses layer-0 features to construct the density map for every frame.

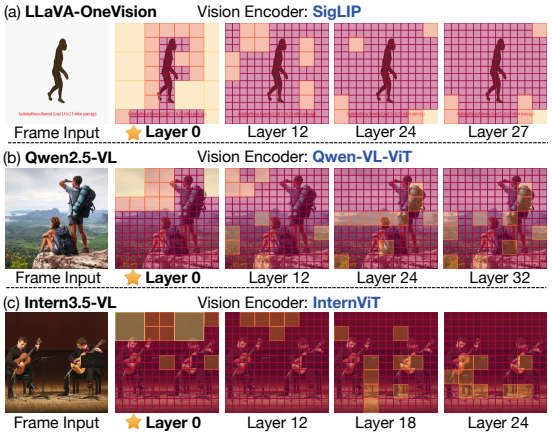


Figure 5: MGDE density maps from three vision encoders. Layer-0 features preserve texture and boundary cues, yielding regional hierarchies, whereas deeper features obscure spatial details.

Density map and token construction. After padding, MGDE tiles each frame with non-overlapping 8×8 roots following the $8 \to 4 \to 2 \to 1$ hierarchy. It splits the highest-gain eligible region when $G_t(R) > \gamma$ while keeping at most $L_{\max}$ leaves across the forest $\mathcal{T}_t$, whose terminal regions form $\mathcal{L}_t$. MGDE then assigns each terminal region $R \in \mathcal{L}_t$ an operational density level ac-

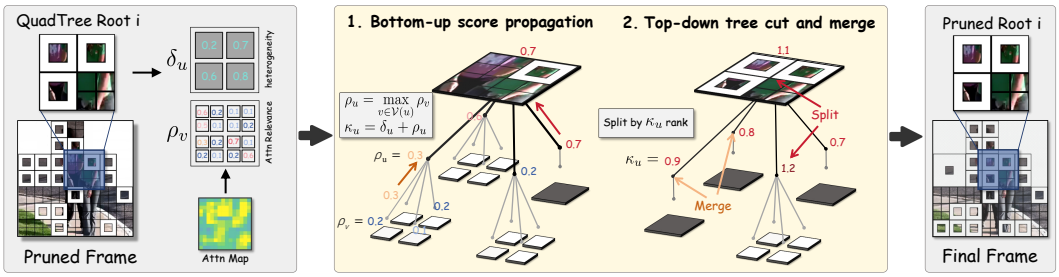


Figure 6: Visualization of ATP on one pruned video frame after MGDE. MGDE provides the surviving visual units and their inherited quadtree. ATP assigns text-to-vision attention scores to these units, propagates them to ancestor nodes, and performs top-down tree cutting and merging.

cording to its side length $s(R)$: regions with $s(R) \geq 4$ receive DL_1, 2×2 regions receive DL_2, and single-cell regions receive DL_3. These labels form the derived density map $\mathcal{M}_t$ and determine token construction: DL_1 regions are dropped, projected features within each DL_2 region are averaged into one token, and each DL_3 cell is kept. Every resulting unit retains its source support, ancestry, and earliest raster position. The units, collected as $\mathcal{V}$ across frames, remain linked to $\mathcal{T}_t$; pooling stays within each frame and the original temporal order is preserved. ATP receives $\mathcal{V}$ and the restriction of $\mathcal{T}_t$ to branches with surviving units for subsequent query-guided pruning within the decoder.

2.4 Attention Tree Pruning

Motivation. After MGDE, the surviving units retain a spatial hierarchy but can still contain content unrelated to the query. To examine how attention-based pruning distributes its selections, we run LearnPruner and ATP on Video-MME at the same visual-token retention ratio using the original query associated with each sample, then count how often the token at each spatial position is retained across frames. Fig. 7 reports position-wise selection frequencies and heatmaps. Under Attention Drift, LearnPruner's flat attention top-k tends to retain peripheral tokens, producing scattered peaks that can fragment relevant visual evidence. In contrast, ATP propagates the same attention scores through the inherited tree and selects spatially coherent branches, shifting retention toward central regions where the primary visual content of a frame is more frequently located. In this way, the tree structure lets ATP evaluate each branch within its multiscale spatial context, which is cheaper and more robust during pruning.

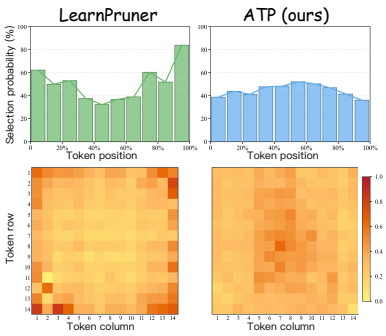


Figure 7: Token-selection frequencies of LearnPruner's top-k pruning and ATP. ATP reduces peripheral selections and retains more tokens in the central region.

Bottom-up score propagation. Let $\mathcal{Q}$ be the question-token positions, H_ℓ the number of heads at decoder layer ℓ, and A^{ℓ}_{qhv} the causal attention probability from question token $q \in \mathcal{Q}$ to visual unit $v \in \mathcal{V}$ in head h. ATP averages these probabilities over question tokens and heads to obtain a_v, then rescales a_v within each frame between its empirical second and 98th percentiles and clips to $[0, 1]$ to obtain the leaf relevance ρ_v. Each surviving unit is associated with the corresponding terminal node of the inherited MGDE quadtree, which connects its attention score to a spatial region. For an internal node u, let $\mathcal{V}(u) \subseteq \mathcal{V}$ denote its surviving descendant units. ATP processes the tree from the leaves to the root and assigns each internal node the strongest relevance among its descendants as the explicit regional relevance term that is subsequently consumed by the top-down tree-cut stage:

$$a_v = \frac{1}{|\mathcal{Q}|\, H_\ell} \sum_{q \in \mathcal{Q}} \sum_{h=1}^{H_\ell} A^{\ell}_{qhv}, \qquad \rho_u = \max_{v \in \mathcal{V}(u)} \rho_v. \tag{1}$$

This propagation converts token-level attention into region-level relevance without introducing learned parameters. The max operation assigns a branch the score of its strongest descendant, so a

Table 1: InternVL3.5-8B results at 10%, 30%, and 50% visual-token retention. Bold and underlined values mark the best and second-best compressed results; the gray line provides the baseline result.

Method	Video QA					Video Caption		Avg.
	VideoMME	NExT-QA	MVBench	Valor32k	VNBench	$\text{VDC}_{\text{Acc.}}$	$\text{VDC}_{\text{Score}}$	
InternVL3.5 (16,384 Tokens)	65.67	80.22	72.19	76.63	59.85	40.47	1.90	100.00%
*Retain 8,192 Tokens (↓ **50.0%**)*								
ToMe (ICLR-2023)	62.64	76.48	69.02	73.14	57.18	38.67	1.82	95.52%
FastV (ECCV-2024)	62.05	75.79	68.27	72.51	56.96	38.45	1.89	95.40%
VisionZip (CVPR-2025)	63.26	77.24	69.42	74.10	57.89	39.20	1.81	96.33%
FrameFusion (ICCV-2025)	64.10	78.61	70.36	75.06	58.75	39.72	1.67	96.46%
STTM (ICCV-2025)	65.23	79.75	72.03	**76.44**	59.76	40.46	1.85	99.35%
FlashVID (ICLR-2026)	**65.70**	80.33	72.09	76.34	59.95	40.67	1.89	99.97%
LearnPruner (ICLR-2026)	64.87	78.91	71.30	75.48	59.19	39.70	1.83	98.25%
FoveaTree (Ours)	65.41	**80.53**	**72.25**	76.26	**60.00**	**41.41**	**1.95**	**100.69%**
*Retain 4,928 Tokens (↓ **70.0%**)*								
ToMe (ICLR-2023)	61.59	75.36	67.60	71.68	56.26	37.70	1.79	93.75%
FastV (ECCV-2024)	61.22	74.57	66.93	70.97	55.45	37.68	1.80	93.14%
VisionZip (CVPR-2025)	62.56	76.80	69.20	73.42	57.33	38.46	1.81	95.54%
FrameFusion (ICCV-2025)	62.24	76.23	68.37	72.57	57.01	38.27	1.77	94.60%
STTM (ICCV-2025)	64.33	79.07	70.96	75.44	58.90	39.57	1.89	98.42%
FlashVID (ICLR-2026)	65.12	**80.06**	71.78	76.28	**59.64**	40.23	1.89	99.50%
LearnPruner (ICLR-2026)	63.66	77.76	69.91	74.52	58.13	39.11	1.83	96.86%
FoveaTree (Ours)	**66.01**	80.04	**72.49**	**76.34**	59.33	**41.39**	**1.95**	**100.62%**
*Retain 1,664 Tokens (↓ **90.0%**)*								
ToMe (ICLR-2023)	56.92	70.01	62.77	66.60	51.88	35.22	1.67	87.06%
FastV (ECCV-2024)	58.22	71.13	63.98	67.94	53.02	35.84	1.67	88.52%
VisionZip (CVPR-2025)	60.10	72.99	65.91	69.76	54.56	36.75	1.75	91.27%
FrameFusion (ICCV-2025)	60.42	73.71	66.36	70.36	55.04	37.33	1.79	92.29%
STTM (ICCV-2025)	63.58	77.92	70.20	74.21	57.82	39.33	1.88	97.25%
FlashVID (ICLR-2026)	63.96	78.27	70.76	74.75	58.35	39.36	1.83	97.37%
LearnPruner (ICLR-2026)	62.87	76.53	68.78	73.13	57.36	38.88	1.77	95.27%
FoveaTree (Ours)	**65.41**	**79.98**	**72.04**	**75.96**	**59.25**	**40.02**	**1.89**	**99.37%**

locally high score is evaluated together with the descendant's ancestral region. The top-down stage uses these propagated scores to determine the resolution retained for each branch under budget.

Top-down tree cut and merge. ATP combines propagated relevance with the heterogeneity of hidden states $h_v^\ell \in \mathbb{R}^d$ within each node. With $n_u = |\mathcal{V}(u)|$, it computes the following scores:

$$\delta_u = 1 - \frac{1}{n_u} \left\| \sum_{v \in \mathcal{V}(u)} \frac{h_v^\ell}{\|h_v^\ell\|_2 + \epsilon} \right\|_2, \qquad \kappa_u = \delta_u + \rho_u. \tag{2}$$

The score κ_u determines the split priority used to construct a budget-constrained frontier for each frame. If MGDE produces M_t units for frame t, ATP targets $K_t = \min(B, M_t)$ outputs. It expands the tree in priority order and merges the residual siblings when a complete expansion would exceed the remaining budget, thereby returning exactly K_t units. For a merged cluster $C \subseteq \mathcal{V}$, ATP attention-weights its hidden states and anchors the result at the highest-attention constituent's original RoPE position. After frame and raster reordering, preceding-layer caches remain unchanged, while subsequent layers populate fresh caches for the compressed prefix. The supplementary material further details the budget control and its boundary handling for exact per-frame output counts.

3 Experiments

3.1 Experimental Setup

Benchmarks. We evaluate on six public video benchmarks spanning short- and long-video understanding, causal and temporal reasoning, and detailed captioning: Video-MME (Fu et al., 2025a), NExT-QA (Xiao et al., 2021), MVBench (Li et al., 2024a), Valor32k (Liu et al., 2025), VNBench (Zhao et al., 2025b), and VDC (Chai et al., 2025). Together, these tasks cover a broad and complementary set of multimodal understanding capabilities, including perception, causal and temporal reasoning, audio-visual association, long-video comprehension, and fine-grained generation. This ensures that pruning-induced degradation is measured across capability profiles rather than on any single skill. Benchmark details and evaluation protocols are provided in the Appendix.

Table 2: Qwen3-VL-8B results at 10%, 30%, and 50% visual-token retention. Bold and underlined values mark the best and second-best compressed results; the gray line provides the baseline result.

Method	Video QA					Video Caption		Avg.
	VideoMME	NExT-QA	MVBench	Valor32k	VNBench	$VDC_{Acc.}$	VDC_{Score}	
Qwen3-VL (5,753 Tokens)	66.15	83.43	67.49	78.27	58.63	45.28	2.31	100.00%
*Retain 2,876 Tokens (↓ **50.0%**)*								
ToMe (ICLR-2023)	62.88	79.27	64.32	74.46	55.83	43.11	2.20	95.17%
FastV (ECCV-2024)	63.59	80.20	64.92	75.35	56.77	43.76	2.24	96.45%
VisionZip (CVPR-2025)	64.25	81.00	65.42	76.30	57.18	44.20	2.24	97.25%
FrameFusion (ICCV-2025)	64.26	81.36	65.45	76.30	57.28	44.21	2.21	97.16%
STTM (ICCV-2025)	65.52	82.37	66.87	77.54	58.14	44.94	2.29	99.07%
FlashVID (ICLR-2026)	65.53	82.83	66.72	77.52	58.16	**45.05**	2.27	99.03%
LearnPruner (ICLR-2026)	65.15	81.83	66.48	76.86	57.81	44.31	**2.30**	98.47%
FoveaTree (Ours)	**65.67**	**83.73**	**67.42**	**78.04**	**58.59**	45.02	**2.30**	**99.74%**
*Retain 1,725 Tokens (↓ **70.0%**)*								
ToMe (ICLR-2023)	61.70	77.94	62.84	72.81	54.81	41.97	2.16	93.22%
FastV (ECCV-2024)	62.43	78.52	63.35	73.40	55.00	42.67	2.20	94.20%
VisionZip (CVPR-2025)	62.76	79.54	64.45	74.68	55.94	42.87	2.19	95.15%
FrameFusion (ICCV-2025)	62.18	78.63	63.37	73.52	55.39	42.49	2.14	93.86%
STTM (ICCV-2025)	64.54	81.90	66.07	76.75	57.47	44.13	**2.29**	98.04%
FlashVID (ICLR-2026)	64.77	82.20	66.25	76.92	57.69	44.44	2.27	98.24%
LearnPruner (ICLR-2026)	63.86	80.54	65.08	75.80	56.71	43.60	2.22	96.50%
FoveaTree (Ours)	**65.38**	**83.45**	**67.37**	**77.79**	**58.30**	**44.97**	**2.29**	**99.42%**
*Retain 575 Tokens (↓ **90.0%**)*								
ToMe (ICLR-2023)	56.00	71.12	57.31	66.45	49.64	38.50	1.98	85.02%
FastV (ECCV-2024)	54.13	68.29	55.20	64.06	47.93	37.02	1.88	81.74%
VisionZip (CVPR-2025)	57.76	72.42	58.79	67.96	50.99	39.24	2.02	87.02%
FrameFusion (ICCV-2025)	59.64	75.12	60.80	70.43	52.84	40.91	2.12	90.36%
STTM (ICCV-2025)	63.38	80.18	64.95	75.00	56.04	43.54	2.27	96.28%
FlashVID (ICLR-2026)	63.95	80.81	65.68	75.79	56.73	43.73	2.21	96.67%
LearnPruner (ICLR-2026)	61.00	76.66	61.92	71.94	54.13	41.88	2.08	91.80%
FoveaTree (Ours)	**64.92**	**82.58**	**67.03**	**77.45**	**57.97**	**44.69**	**2.28**	**98.81%**

Implementation Details. We evaluate InternVL3.5-8B and Qwen3-VL-8B with identical inputs and decoding settings, and compare ToMe (Bolya et al., 2023), FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025), FrameFusion (Fu et al., 2025b), STTM (Hyun et al., 2025), FlashVID (Fan et al., 2026), and LearnPruner (Takezoe et al., 2026). The proposed MGDE module uses layer-0 features, a root side length of 8, a per-frame leaf cap, and a merge-gain threshold of 0.005. The proposed ATP module is inserted at decoder layer 18. We run batch-one bfloat16 inference with SDPA and greedy decoding on one NVIDIA A800 80 GB GPU, synchronize CUDA timing, and exclude warm-up. Efficiency is measured on Video-MME with at most 64 frames. Exact retained-token counts and results on additional backbones are provided in the Appendix, together with full implementation details for all evaluated models, retention configurations, and benchmarks.

3.2 Comparison with State-of-the-Art Methods

Tables 1 and 2 report all seven metrics at 50%, 30%, and 10% visual-token retention on both evaluated backbones. The five Video QA benchmarks use answer accuracy, while VDC reports Acc. and Score. Avg. (%) equally averages each reported metric relative to its corresponding Full result. FoveaTree consistently achieves the highest Avg. for every reported setting on both backbones, and the margin over the strongest competing baseline widens as the token budget decreases.

Results on InternVL3.5-8B. On InternVL3.5-8B, FoveaTree obtains 100.62%, 100.69%, and 99.37% Avg. at 30%, 50%, and 10% retention, respectively. At 30%, it leads on five of the seven metrics, including 66.01 on Video-MME and 41.39/1.95 on $VDC_{Acc.}$/VDC_{Score}, and ranks a close second to the strongest baseline on the two remaining metrics of NExT-QA and VNBench.

Results on Qwen3-VL-8B. On Qwen3-VL-8B, FoveaTree obtains 99.42%, 99.74%, and 98.81% Avg. at 30%, 50%, and 10% retention, respectively, and leads on all seven metrics at 30%, including 65.38 on Video-MME and 44.97/2.29 on $VDC_{Acc.}$/VDC_{Score}. Across both backbones, FoveaTree remains ahead of the compared methods at every budget, and its gains extend to VDC detailed captioning, which requires accurate descriptions of objects, events, and visual details across frames.

Table 3: Efficiency analysis at 30% and 10% visual-token retention on InternVL3.5 on Video-MME.

Method	Time	Mem	Latency	TFLOPs	Acc.	Time	Mem	Latency	TFLOPs	Acc.
InternVL3.5 (16,384 Tokens)	111:47	20.26	2.484	322.0	65.67	111:47	20.26	2.484	322.0	65.67
	*Retain 4,928 Tokens (↓ **70.0%**)*					*Retain 1,664 Tokens (↓ **90.0%**)*				
ToMe (ICLR-2023)	74:58	17.67	1.666	86.5	61.59	89:28	16.90	1.988	33.6	56.92
FastV (ECCV-2024)	106:34	19.09	2.368	204.2	61.22	98:06	19.09	2.180	177.8	58.22
VisionZip (CVPR-2025)	49:11	17.67	1.093	86.5	62.56	44:25	16.90	0.987	33.6	60.10
FrameFusion (ICCV-2025)	54:22	17.94	1.208	93.0	62.24	35:52	17.94	0.797	43.9	60.42
STTM (ICCV-2025)	82:53	19.34	1.842	204.2	64.33	86:02	19.15	1.912	177.8	63.58
FlashVID (ICLR-2026)	95:16	**17.95**	2.117	197.3	65.12	66:12	**17.00**	1.471	136.7	63.96
LearnPruner (ICLR-2026)	92:10	19.74	2.048	**143.2**	63.66	69:56	18.92	1.554	132.9	62.87
FoveaTree (Ours)	**90:16**	19.36	**2.006**	150.8	**66.01**	**65:07**	19.36	**1.447**	**125.1**	**65.41**

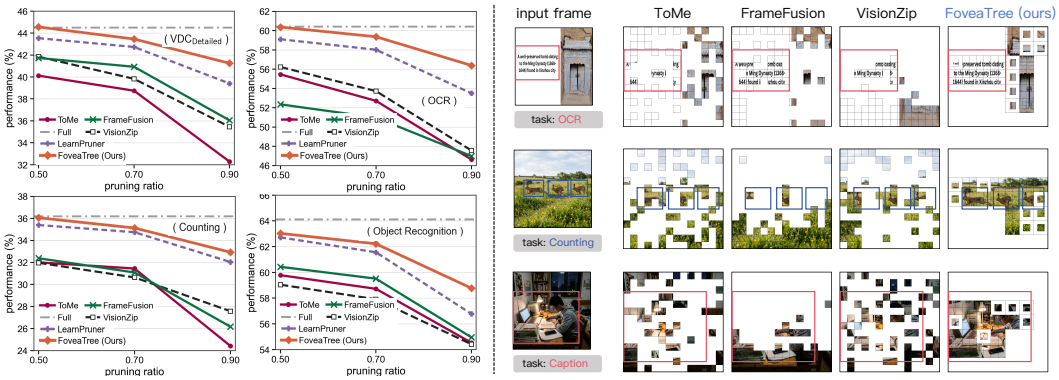


Figure 8: Fine-grained performance and retained-token visualizations. FoveaTree achieves the strongest results across the shown token-retention ratios, while the right panel illustrates that it preserves relevant regions more completely than the compared methods under matched budgets.

3.3 Efficiency Analysis

To measure the efficiency of FoveaTree, we record total inference time, peak memory, latency, and TFLOPs on Video-MME together with Video-MME accuracy from the complete benchmark evaluation, as reported in Table 3. At 30% retention on InternVL3.5-8B, FoveaTree reduces TFLOPs by 53.2% relative to Full (322.0 to 150.8) with a 1.24× latency speedup (2.484 s to 2.006 s), while peak memory stays within 4.4% of the unpruned model (20.26 to 19.36 GiB). Relative to the earlier pruning methods (shown in gray), FoveaTree leads in Video-MME accuracy by a large margin; relative to the more recent and stronger baselines, it attains the lowest inference time at both budgets while remaining the most accurate, demonstrating the superior accuracy-efficiency trade-off.

3.4 Ablation Study

We conduct ablations on Video-MME and VDC using InternVL3.5-8B. Configurations containing ATP use a nominal 30% per-frame target, whereas MGDE-only reports its adaptive output before ATP budget refinement. Avg. equally weights the two reported metrics after each is normalized by the corresponding Full result of the unpruned InternVL3.5-8B model on the very same inputs.

Component effectiveness. Table 4 evaluates MGDE and ATP while recording the latency of each configuration. The first row retains the original visual sequence. MGDE-only constructs the quadtree and performs density-guided token operations before prefill without ATP. ATP-only applies query-guided pruning to a complete per-frame quadtree whose leaves are the original projected tokens. Combining both stages reaches 101.40% Avg. under the nominal 30% target, improving Avg. by 0.78 and 0.92 points over MGDE-only and ATP-only while reducing latency to 2.006 s and preserving complementary regional and query-guided gains across both stages.

Layer selection. Under the nominal 30% target, we study the feature and decoder layers used by both stages. Table 5 shows that layer-0 vision features give MGDE the best encoder-layer result, reaching 101.40% Avg. Table 6 shows that ATP at decoder layer 18 gives the highest Avg. among

Table 4: Component ablation of MGDE and ATP.

MGDE	ATP	VideoMME	$VDC_{Acc.}$	Avg.	Lat.
×	×	65.67	40.47	100.00%	2.484
✓	×	65.50	41.08	100.62%	2.247
×	✓	65.44	41.00	100.48%	2.086
✓	✓	**66.01**	**41.39**	**101.40%**	**2.006**

Table 5: MGDE feature-layer study.

Layer	VideoMME	$VDC_{Acc.}$	Avg.
0	**66.01**	**41.39**	**101.40%**
12	65.48	40.99	100.50%
24	65.18	40.78	100.01%
32	62.55	38.65	95.37%

Table 6: ATP insertion-layer study.

Layer	VideoMME	$VDC_{Acc.}$	Avg.
6	65.29	40.84	100.17%
12	65.43	40.97	100.43%
18	**66.01**	**41.39**	**101.40%**
24	65.88	41.27	101.15%

Table 7: ATP scoring and selection strategies.

Strategy	VideoMME	$VDC_{Acc.}$	Avg.	Lat.
Flat Top-k	65.05	40.62	99.71%	**1.896**
Debiased Top-k	65.39	40.93	100.36%	2.700
Debiased ATP	**66.11**	41.39	**101.47%**	2.813
ATP (Ours)	66.01	**41.39**	101.40%	2.006



Figure 9: Qualitative comparison of LearnPruner and FoveaTree with video frames. The examples illustrate that FoveaTree preserves key objects and their surrounding regions more completely.

the tested layers, also reaching 101.40%. We use layers 0 and 18 for regional analysis and for query-guided pruning in all subsequent experiments and qualitative visualizations.

ATP strategy. Table 7 compares ATP with flat Top-k and debiased variants at 30% retention, reporting both Avg. and latency. Following representation-difference calibration (Liu et al., 2024c; Zhao et al., 2025a), debiasing subtracts layer-18 hidden states obtained from an all-zero reference video under the same query before recomputing text-to-vision attention, thereby reducing video-independent bias. ATP reaches 101.40% Avg. and 2.006 s, within 0.07 points of Debiased ATP while avoiding its additional reference pass at 2.813 s, providing direct evidence of stronger robustness to Attention Drift at substantially lower latency than every debiased alternative in this comparison.

3.5 Visualization

Figs. 8 and 9 visualize FoveaTree with respect to the two challenges introduced in the Introduction. Fig. 8 reports fine-grained performance across retention ratios, where FoveaTree consistently outperforms similarity-based methods, and its right panel shows more complete preservation of regional evidence, supporting hierarchy-aware pruning. Fig. 9 presents two Video-MME examples in which, under the same token budget, FoveaTree is less affected by Attention Drift than LearnPruner and more accurately preserves query-relevant core regions and their surrounding evidence intact.

4 Conclusion

We presented FoveaTree, a training-free framework motivated by Hierarchy-blind Pruning and Attention Drift in Video LLMs. MGDE reduces redundant visual units through a compressibility-derived spatial hierarchy, while ATP propagates query-conditioned attention through the surviving tree. Experiments on two Video LLMs and six benchmarks at three retention budgets demonstrate stronger performance retention with competitive efficiency, establishing regional hierarchy as a practical bridge between query-independent compression and query-aware selection. The framework is more complex than basic pruning methods and lacks explicit temporal processing; future work will simplify the pipeline and extend the hierarchy across time to model motion and event dynamics.

Ethics Statement

This work does not involve human participants, human subjects, personally identifiable information, or the collection of new personal data. We evaluate on publicly released video benchmarks, including Video-MME, NExT-QA, MVBench, Valor32k, VNBench, and VDC, following their intended evaluation protocols and applicable usage terms. The proposed method is an inference-time visual-token pruning framework and does not generate new training data or modify the parameters of the evaluated Video LLMs. As with other video-language systems, the evaluated models may inherit biases, hallucinations, and errors from their underlying data and pretrained models. Token pruning may additionally remove visual evidence in difficult cases, particularly for fine-grained, temporal, or safety-critical content. We therefore view FoveaTree as an efficiency technique rather than a substitute for human oversight in high-stakes applications.

Reproducibility Statement

We evaluate FoveaTree on four publicly available Video LLM backbones, six public video benchmarks, and three visual-token retention budgets. The paper and supplementary material describe the MGDE and ATP algorithms, regional distortion and attention-score formulations, quadtree construction, budget resolution procedure, model configurations, benchmark protocols, decoding settings, hardware, timing methodology, and token-budget audits. The supplementary material also provides additional backbone results, component and layer ablations, efficiency analyses, and implementation details for the compared baselines. Our code, configuration files, and evaluation scripts will be released in the final version of the paper to ensure that all reported results can be reproduced.

AI Use Statement

Generative AI tools assisted with translation, manuscript drafting, literature search, reference formatting, LaTeX typesetting, and plotting code. They provided feedback on methodological descriptions and experimental analyses, assisted with mathematical derivations and their exposition, and supported numerical calculations when preparing results. The authors retain responsibility for the final manuscript, including all claims, references, numerical results, and AI-assisted artifacts.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, et al. Flamingo: a Visual Language Model for Few-Shot Learning. In *NeurIPS*, 2022.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. HiRED: Attention-Guided Token Dropping for Efficient Inference of High-Resolution Vision-Language Models. In *AAAI*, 2025.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, et al. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/abs/2308.12966.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. MADTP: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *CVPR*, pp. 15710–15720, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021.

Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. AuroraCap: Efficient, Performant Video Detailed Captioning and a New Benchmark. In *ICLR*, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, pp. 19–35, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. In *ECCV*, 2024b.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, et al. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *Science China Information Sciences*, 2024c.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, et al. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. In *CVPR*, 2024d.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the Throttle: Revisiting Visual Token Pruning for Vision-Language Model Acceleration. In *ICCV*, 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 *ICLR*, 2026. URL https://github.com/Fanziyang-v/FlashVID.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive Token Sampling For Efficient Vision Transformers. In *ECCV*, 2022.

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 *CVPR*, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining similarity and importance for video token reduction on large vision language models. In *ICCV*, 2025b.

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Donglin Wang, Honggang Chen, Qingsen Yan, et al. Filter, Correlate, Compress: Training-Free Token Reduction for MLLM Acceleration. In *AAAI*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*, 2022.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. ZipVL: Efficient Large Vision-Language Models with Dynamic Token Sparsification. *arXiv preprint arXiv:2410.08584*, 2024. URL https://arxiv.org/abs/2410.08584.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video LLMs. In *ICCV*, 2025. doi: 10.1109/ICCV51701.2025.02224.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-UniVi: Unified Visual Representation Empowers Large Language Models with Image and Video Understanding. In *CVPR*, 2024.

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.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, et al. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. In *CVPR*, 2024a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models. In *ECCV*, 2024b.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-Gemini: Mining the Potential of Multi-modality Vision Language Models. *TPAMI*, 2026.

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

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 *EMNLP*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning. In *CVPR*, 2024a.

Jing Liu, Sihan Chen, Xingjian He, Longteng Guo, Xinxin Zhu, Weining Wang, and Jinhui Tang. VALOR: Vision-Audio-Language Omni-Perception Pretraining Model and Dataset. *TPAMI*, 2025.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *TACL*, 2024b.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. Reducing hallucinations in vision-language models via latent space steering. *arXiv preprint arXiv:2410.15778*, 2024c.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, et al. MMBench: Is Your Multi-modal Model an All-around Player? In *ECCV*, 2024d.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 2021.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In *ACL*, 2024.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token Pooling in Vision Transformers. In *WACV*, 2023.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards Interpreting Visual Information Processing in Vision-Language Models. In *ICLR*, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *NeurIPS*, 2021.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video Understanding. In *CVPR*, 2024.

Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: What Can 8 Learned Tokens Do for Images and Videos? In *NeurIPS*, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *ICCV*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Rinyoichi Takezoe, Yaqian Li, Zi-Hao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. LearnPruner: Rethinking attention-based token pruning in vision language models. In *ICLR*, 2026.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs. In *CVPR*, 2024.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and Modality-Aware KV Cache Compression for Vision-Language Model Inference Acceleration. In *ICLR*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, et al. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*, 2024a. URL https://arxiv.org/abs/2409.12191.

Qiang Wang, Xinyuan Gao, Yuhang He, Jizhou Han, Jiangyang Li, Songlin Dong, Zhiheng Ma, and Yihong Gong. VDC-Agent: When video detailed captioners evolve themselves via agentic self-reflection. In *ECCV*, 2026. doi: 10.1007/978-3-032-37592-6_30.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, et al. InternVideo2: Scaling Foundation Models for Multimodal Video Understanding. In *ECCV*, 2024b.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop Looking for Important Tokens in Multimodal Language Models: Duplication Matters More. In *EMNLP*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *ICLR*, 2024.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, et al. PyramidDrop: Accelerating Your Large Vision-Language Models via Pyramid Visual Redundancy Reduction. In *CVPR*, 2025.

Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, et al. LLaVA-UHD: an LMM Perceiving Any Aspect Ratio and High-Resolution Images. In *ECCV*, 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 *CVPR*, pp. 19792–19802, 2025.

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. DeCo: Decoupling Token Compression from Semantic Abstraction in Multimodal Large Language Models. *arXiv preprint arXiv:2405.20985*, 2024. URL https://arxiv.org/abs/2405.20985.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. ATP-LLaVA: Adaptive token pruning for large vision language models. In *CVPR*, pp. 24972–24981, 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities. In *ICML*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023.

Evelyn Zhang, Fufu Yu, Aoqi Wu, Zichen Wen, Ke Yan, Shouhong Ding, Biqing Qi, and Linfeng Zhang. D^2Pruner: Debiased importance and structural diversity for MLLM token pruning. In *AAAI*, 2026.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, et al. Beyond Text-Visual Attention: Exploiting Visual Cues for Effective Token Pruning in VLMs. In *ICCV*, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, et al. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *ICML*, 2025b.

Kai Zhao, Wubang Yuan, Yuchen Lin, Liting Ruan, Xiaofeng Lu, Deng-Ping Fan, Ming-Ming Cheng, and Dan Zeng. Attention debiasing for token pruning in vision–language models. *arXiv preprint arXiv:2508.17807*, 2025a.

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle In A Video Haystack: A Scalable Synthetic Evaluator for Video MLLMs. In *ICLR*, 2025b.

Yuke Zhu, Chi Xie, Shuang Liang, Bo Zheng, and Sheng Guo. FocusLLaVA: A Coarse-to-Fine Approach for Efficient and Effective Visual Token Compression. *arXiv preprint arXiv:2411.14228*, 2024. URL https://arxiv.org/abs/2411.14228.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. Don't Just Chase "Highlighted Tokens" in MLLMs: Revisiting Visual Holistic Context Retention. In *NeurIPS*, 2025.

Supplementary Material

CONTENTS

A More Experimental Results **16**

A.1 Additional Experiments on Qwen2.5-VL 16

A.2 Additional Experiments on LLaVA-OneVision 16

A.3 Additional Analysis 17

B Evaluation Benchmarks **19**

C Implementation Details **20**

C.1 Reproduction Details of Compared Baselines 20

C.2 Reproduction Details of FoveaTree 20

C.3 Token Budget Alignment 20

C.4 Properties of Tree-Based Aggregation 22

D Related Work **23**

E More Visualizations **23**

E.1 Visualizations of MGDE 24

E.2 Visualizations of ATP 25

E.3 Qualitative Analysis on InternVL3.5 25

E.4 Qualitative Analysis on Qwen3-VL 26

Table 8: Qwen2.5-VL-7B results at 10%, 30%, and 50% visual-token retention. Bold and underlined values mark the best and second-best compressed results; Full provides the unpruned reference.

Method	Video QA					Video Caption		Avg.
	VideoMME	NExT-QA	MVBench	Valor32k	VNBench	$VDC_{Acc.}$	VDC_{Score}	
Qwen2.5-VL (5,753 Tokens)	61.85	82.32	69.16	76.94	66.32	45.40	2.34	100.00%
*Retain 2,876 Tokens (↓ **50.0%**)*								
ToMe (ICLR-2023)	58.80	78.45	65.73	73.27	63.18	43.44	2.23	95.27%
FastV (ECCV-2024)	59.64	79.42	66.87	74.26	64.16	43.70	2.26	96.53%
VisionZip (CVPR-2025)	60.19	79.95	67.14	74.90	64.65	44.28	2.27	97.27%
FrameFusion (ICCV-2025)	59.76	79.47	66.95	74.31	64.05	43.78	2.26	96.59%
STTM (ICCV-2025)	61.35	81.54	**68.74**	76.29	65.90	44.97	2.32	99.19%
FlashVID (ICLR-2026)	**61.39**	81.63	68.64	76.34	65.50	44.95	2.32	99.11%
LearnPruner (ICLR-2026)	60.46	80.37	67.80	75.17	64.85	44.48	2.29	97.82%
FoveaTree (Ours)	61.30	**82.16**	68.55	**76.66**	**65.91**	**45.17**	**2.33**	**99.45%**
*Retain 1,725 Tokens (↓ **70.0%**)*								
ToMe (ICLR-2023)	56.33	74.92	63.21	70.23	60.60	41.41	2.13	91.20%
FastV (ECCV-2024)	56.74	75.63	63.45	70.71	61.07	41.78	2.15	91.89%
VisionZip (CVPR-2025)	58.28	77.60	65.40	72.74	62.60	42.70	2.21	94.36%
FrameFusion (ICCV-2025)	57.53	76.87	64.59	71.81	61.89	42.50	2.18	93.32%
STTM (ICCV-2025)	60.81	81.08	67.96	75.54	65.21	44.54	2.30	98.28%
FlashVID (ICLR-2026)	60.99	81.50	68.31	76.16	65.63	**44.77**	**2.31**	98.81%
LearnPruner (ICLR-2026)	59.52	79.35	66.51	74.22	63.80	43.70	2.25	96.27%
FoveaTree (Ours)	**61.44**	**82.23**	**68.58**	**76.64**	**65.85**	44.69	2.31	**99.21%**
*Retain 575 Tokens (↓ **90.0%**)*								
ToMe (ICLR-2023)	51.43	68.34	57.62	64.09	55.16	37.67	1.94	83.12%
FastV (ECCV-2024)	51.09	67.66	56.96	63.30	54.66	37.52	1.93	82.42%
VisionZip (CVPR-2025)	57.18	75.97	63.90	71.15	61.36	41.76	2.16	92.35%
FrameFusion (ICCV-2025)	55.72	73.99	62.21	69.30	59.75	40.78	2.11	90.01%
STTM (ICCV-2025)	60.04	79.93	67.37	74.63	64.37	43.97	2.27	97.07%
FlashVID (ICLR-2026)	60.61	80.79	67.65	75.38	65.04	**44.43**	**2.29**	97.96%
LearnPruner (ICLR-2026)	58.11	77.16	64.67	71.94	62.34	42.45	2.19	93.68%
FoveaTree (Ours)	**61.37**	**82.02**	**68.76**	**76.57**	**65.98**	43.40	2.24	**98.37%**

A More Experimental Results

A.1 Additional Experiments on Qwen2.5-VL

The main paper evaluates InternVL3.5-8B and Qwen3-VL-8B. To examine generality beyond these two backbones, this subsection applies the same evaluation protocol to Qwen2.5-VL-7B. Table 8 reports the full seven-metric protocol at 50%, 30%, and 10% retention budgets, following the identical frame sampling and decoding configuration. These results quantify how each pruning method behaves across the QA benchmarks and the captioning score under each budget. Qwen2.5-VL-7B (Bai et al., 2025b) uses a native dynamic-resolution vision transformer with window attention and 2D-RoPE. Its spatial token count varies with input resolution, and MGDE and ATP operate on each temporal unit's native grid.

A.2 Additional Experiments on LLaVA-OneVision

To examine generality beyond the two backbones used in the main comparison, we apply the same evaluation protocol to LLaVA-OneVision-7B (Li et al., 2025). Table 9 reports the complete seven-metric protocol at 50%, 30%, and 10% retention. Frame sampling follows the model's official video configuration, and all methods share identical inputs and decoding settings. FoveaTree maintains competitive performance across all three retention budgets, with particularly strong results on the Video QA benchmarks at 10% retention. Unlike the two main backbones, LLaVA-OneVision-7B pairs a SigLIP-SO400M vision encoder with a Qwen2-7B decoder and encodes each sampled frame (one native temporal unit) into a fixed low-resolution grid of 196 tokens, so the visual sequence is uniform across units rather than resolution-adaptive. MGDE uses the same quadtree construction rule on each unit, and ATP is inserted at decoder layer 18, transferring the protocol without architectural modification.

Table 9: LLaVA-OV-7B results at 10%, 30%, and 50% visual-token retention. Bold and underlined values mark the best and second-best compressed results; Full provides the unpruned reference.

Method	Video QA					Video Caption		Avg.
	VideoMME	NExT-QA	MVBench	Valor32k	VNBench	$VDC_{Acc.}$	VDC_{Score}	
LLaVA-OV (12,544 Tokens)	58.56	80.57	58.26	74.11	62.87	41.85	2.17	100.00%
*Retain 6,272 Tokens (↓ **50.0%**)*								
ToMe (ICLR-2023)	57.17	78.86	56.84	72.48	61.51	41.13	2.12	97.81%
FastV (ECCV-2024)	55.92	76.98	55.79	70.83	60.24	39.88	2.08	95.62%
VisionZip (CVPR-2025)	57.04	78.32	56.59	72.21	61.35	40.86	2.11	97.38%
FrameFusion (ICCV-2025)	56.89	78.21	56.72	71.97	61.05	40.57	2.11	97.14%
STTM (ICCV-2025)	58.07	79.79	57.91	73.47	62.46	41.44	2.15	99.17%
FlashVID (ICLR-2026)	**58.71**	80.76	58.38	74.31	62.66	41.85	**2.17**	100.09%
LearnPruner (ICLR-2026)	57.81	79.43	57.70	73.11	62.08	41.41	2.14	98.76%
FoveaTree (Ours)	58.56	**81.46**	**58.90**	**74.93**	**63.56**	**42.31**	2.16	**100.72%**
*Retain 3,776 Tokens (↓ **70.0%**)*								
ToMe (ICLR-2023)	55.13	75.81	55.06	69.93	59.39	39.47	2.05	94.34%
FastV (ECCV-2024)	54.17	74.66	53.88	68.70	58.40	38.84	2.01	92.67%
VisionZip (CVPR-2025)	56.52	77.80	56.45	71.78	60.79	40.32	2.10	96.66%
FrameFusion (ICCV-2025)	56.44	77.96	56.39	71.67	60.80	40.60	2.10	96.73%
STTM (ICCV-2025)	57.49	79.25	57.16	72.65	61.73	40.99	2.13	98.14%
FlashVID (ICLR-2026)	58.37	80.74	58.12	74.21	62.86	41.71	**2.17**	99.92%
LearnPruner (ICLR-2026)	56.51	77.89	56.17	71.70	60.66	40.40	2.10	96.59%
FoveaTree (Ours)	**58.46**	**81.28**	**58.76**	**74.75**	**63.41**	**42.21**	2.15	**100.46%**
*Retain 1,280 Tokens (↓ **90.0%**)*								
ToMe (ICLR-2023)	49.43	67.90	49.30	62.67	53.08	35.25	1.83	84.41%
FastV (ECCV-2024)	47.96	65.64	47.57	60.44	51.36	34.30	1.77	81.68%
VisionZip (CVPR-2025)	52.10	71.53	51.80	65.94	55.97	37.02	1.93	88.87%
FrameFusion (ICCV-2025)	51.10	70.14	50.75	64.66	54.87	36.40	1.89	87.15%
STTM (ICCV-2025)	56.54	77.82	56.47	71.51	60.70	40.30	2.10	96.60%
FlashVID (ICLR-2026)	57.73	79.64	57.26	73.07	61.97	41.19	**2.14**	98.56%
LearnPruner (ICLR-2026)	54.67	75.02	54.09	68.83	58.72	38.86	2.02	93.08%
FoveaTree (Ours)	**58.26**	**80.58**	**58.27**	**74.12**	**62.88**	**41.86**	2.12	**99.61%**

A.3 Additional Analysis

Efficiency on additional backbones. Table 10 extends the efficiency analysis of the main paper to Qwen3-VL, Qwen2.5-VL, and LLaVA-OneVision. For each backbone, it reports total inference time, peak memory, latency, and TFLOPs together with Video-MME accuracy at 30% and 10% retention, measured under the identical protocol: the same Video-MME inputs, hardware, and timing procedure as the main efficiency analysis.

Table 10: Efficiency analysis at 30% and 10% visual-token retention on the additional backbones on Video-MME.

Method	Time	Mem	Latency	TFLOPs	Acc.	Time	Mem	Latency	TFLOPs	Acc.
Qwen3-VL (5,753 Tokens)	41:24	18.26	0.920	77.247	66.15	41:24	18.26	0.920	77.247	66.15
	*Retain 1,725 Tokens (↓ **70.0%**)*					*Retain 575 Tokens (↓ **90.0%**)*				
ToMe (ICLR-2023)	29:40	**17.35**	0.659	**26.333**	61.70	32:05	**17.05**	0.713	**12.697**	56.00
STTM (ICCV-2025)	31:54	18.02	0.709	51.789	64.54	31:34	17.97	0.702	44.971	63.38
FlashVID (ICLR-2026)	26:57	17.51	0.599	28.835	64.77	**20:59**	17.08	**0.466**	13.469	63.95
FoveaTree (Ours)	**24:25**	17.80	**0.543**	34.303	**65.38**	22:40	17.62	0.504	27.835	**64.92**
Qwen2.5-VL (5,753 Tokens)	27:39	16.83	0.615	83.566	61.85	27:39	16.83	0.615	83.566	61.85
	*Retain 1,725 Tokens (↓ **70.0%**)*					*Retain 575 Tokens (↓ **90.0%**)*				
ToMe (ICLR-2023)	22:14	**16.12**	0.494	**24.708**	56.33	23:52	**15.98**	0.530	**9.089**	51.43
STTM (ICCV-2025)	25:41	16.72	0.571	62.544	60.81	24:05	16.72	0.535	56.966	60.04
FlashVID (ICLR-2026)	23:31	16.17	0.522	28.776	60.99	**16:18**	**15.98**	**0.362**	10.519	60.61
FoveaTree (Ours)	**20:23**	16.65	**0.453**	41.437	**61.44**	19:14	16.65	0.427	35.897	**61.37**
LLaVA-OV (12,544 Tokens)	53:33	17.88	1.190	197.331	58.56	53:33	17.88	1.190	197.331	58.56
	*Retain 3,776 Tokens (↓ **70.0%**)*					*Retain 1,280 Tokens (↓ **90.0%**)*				
ToMe (ICLR-2023)	37:01	**16.33**	0.823	**53.796**	55.13	37:57	**15.87**	0.843	**18.576**	49.43
STTM (ICCV-2025)	46:59	17.64	1.044	146.065	57.49	46:44	17.64	1.039	133.487	56.54
FlashVID (ICLR-2026)	33:51	16.47	0.752	63.186	58.37	**18:51**	16.08	**0.419**	21.631	57.73
FoveaTree (Ours)	**31:01**	17.77	**0.689**	91.245	**58.46**	28:22	17.77	0.630	80.568	**58.26**

Table 11: Ablation on the maximum quadtree depth D at 30% retention. Latency follows the efficiency protocol of the main paper.

Depth D	VideoMME	$\text{VDC}_{\text{Acc.}}$	Avg.	Lat.
$D = 2$	63.87	38.92	96.71%	**1.871**
$D = 3$	65.24	40.31	99.47%	1.942
$D = 4$	**66.01**	**41.39**	**101.40%**	2.006
$D = 5$	65.83	41.02	100.80%	2.089

Quadtree depth. These supplementary ablations are conducted on the InternVL3.5-8B backbone, i.e., the primary model of the main paper. The depth setting controls the MGDE hierarchy and the spatial extent of its root regions. This ablation varies the maximum depth $D \in \{2, 3, 4, 5\}$, where $D = 4$ corresponds to the default $8 \to 4 \to 2 \to 1$ schedule, while $D = 5$ starts from a single 16×16 root covering the entire frame grid and follows the $16 \to 8 \to 4 \to 2 \to 1$ schedule. Table 11 reports Video-MME accuracy, VDC accuracy, and the average relative performance, together with latency measured under the efficiency protocol of the main paper.

Table 12: Ablation on the ATP split-score composition at 30% retention.

κ_u	VideoMME	$\text{VDC}_{\text{Acc.}}$	Avg.
$\kappa_u = \delta_u$	65.18	40.42	99.57%
$\kappa_u = \rho_u$	64.15	39.47	97.61%
$\kappa_u = a_u$	63.36	38.71	96.07%
$\kappa_u = \delta_u + \rho_u$	**66.01**	**41.39**	**101.40%**

Score composition. The ATP split priority combines feature heterogeneity with propagated attention relevance, $\kappa_u = \delta_u + \rho_u$. This ablation isolates the two terms: ranking nodes by heterogeneity alone ($\kappa_u = \delta_u$) preserves content-diverse regions but ignores the query, whereas ranking by relevance alone ($\kappa_u = \rho_u$) follows attention without accounting for the internal diversity of a region. We further include the uncalibrated variant $\kappa_u = a_u$, where $a_u = \max_{v \in \mathcal{V}(u)} a_v$ propagates the raw averaged leaf attention defined in the main paper. This bypasses the within-unit percentile rescaling and isolates the contribution of that calibration step. Table 12 compares the four configurations under the same retention budget. The combined score retains the strongest results on both metrics, indicating that the two terms are complementary: relevance locates query-related branches while heterogeneity prevents merging internally diverse regions.

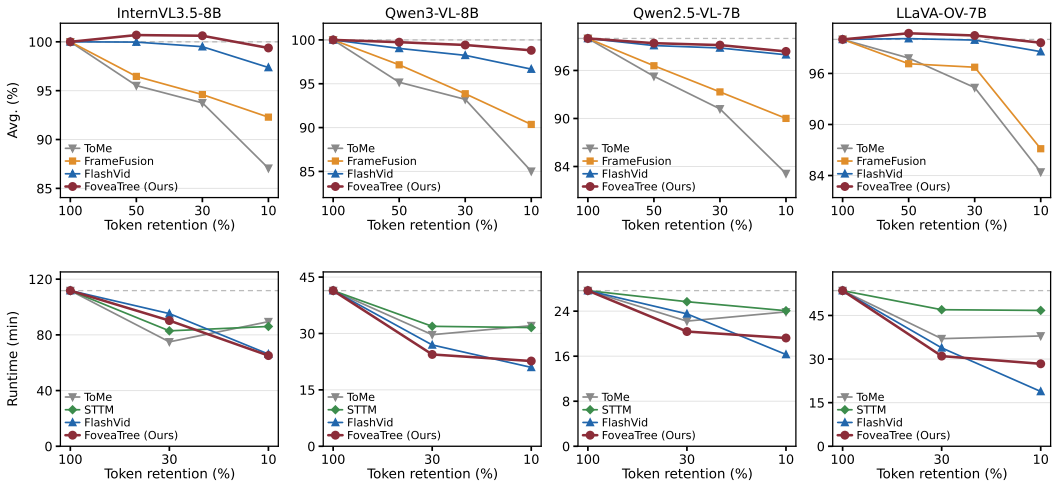


Figure 10: Cross-backbone comparison. Top: average relative performance versus the visual-token retention ratio on InternVL3.5-8B, Qwen3-VL-8B, Qwen2.5-VL-7B, and LLaVA-OneVision-7B for FoveaTree, FlashVID, FrameFusion, and ToMe; the dashed line marks the unpruned reference. Bottom: total Video-MME runtime under the efficiency protocol of the main paper for FoveaTree, FlashVID, STTM, and ToMe; the dashed line marks the runtime of the unpruned model.

Cross-backbone result analysis. Figure 10 summarizes performance retention and runtime across four backbones. FoveaTree maintains strong performance across these architectures while reducing inference time relative to the unpruned models. At 30% retention, it achieves both the highest Video-MME accuracy and the lowest latency among the methods in Table 10 on Qwen3-VL, Qwen2.5-VL, and LLaVA-OneVision. On InternVL3.5, it is more accurate and faster than FlashVID and

LearnPruner. These results demonstrate a favorable accuracy–latency trade-off across backbones. At 10% retention, it preserves the highest Video-MME accuracy among these methods, while FlashVID is faster on the Qwen and LLaVA backbones.

The results also suggest that backbones with larger original visual-token sequences can offer more room for compression. This is particularly evident on InternVL3.5, whose 16,384-token input can be compressed at both 50% and 30% retention while exceeding the unpruned reference in average relative performance. These gains show that retaining more visual tokens does not necessarily improve understanding. We hypothesize that excessive visual redundancy, together with a model's limited ability to use long visual contexts effectively, can reduce the benefit of additional tokens; structured compression may alleviate this burden by retaining informative regions. The cross-backbone observations are consistent with this explanation, although differences in architecture and preprocessing prevent attributing the effect to token count alone.

B Evaluation Benchmarks

We use the official test splits, subject to the subset specifications below.

Video-MME. Video-MME (Fu et al., 2025a) evaluates video understanding across short, medium, and long videos. Its questions cover perception and reasoning in diverse content domains. We evaluate without subtitles, report answer accuracy, and use the dataset for efficiency measurements with a 64-frame cap. The fine-grained analysis examines OCR, counting, and object-related questions, which place different demands on the retained visual content.

NExT-QA. NExT-QA (Xiao et al., 2021) includes descriptive, temporal, and causal questions about video events. In addition to recognizing the visible content, the model must connect actions and explain their temporal or causal relationships. We report question-answering accuracy to assess whether token pruning preserves the evidence required for these judgments.

MVBench. MVBench (Li et al., 2024a) organizes video understanding into multiple tasks involving actions, objects, spatial relationships, and temporal changes. Its diversity complements the broad evaluation in Video-MME. We use the benchmark's accuracy metric and retain the original evaluation protocol when comparing the full model and compressed models. The official release contains 4,000 samples, but 200 samples of the Fine-grained Pose task are missing from the version we evaluate, so all reported numbers use the remaining 3,800 samples. Every method, including the uncompressed reference, is evaluated on the identical 3,800-sample subset, ensuring a fair comparison.

Valor32k. The VALOR dataset (Liu et al., 2025) supports vision, audio, and language understanding. We evaluate on the visual subset of Valor32k using the visual question-answering setting. This setting measures the model's ability to answer questions using video content and complements the action-oriented evaluation in NExT-QA and MVBench.

VNBench. VNBench (Zhao et al., 2025b) evaluates retrieval, ordering, and counting through the Video Needle In A Haystack construction. Relevant visual events are embedded within longer sequences, making the benchmark useful for assessing temporal localization and the retention of sparse evidence. We report accuracy under the same input configuration for the full and compressed models.

VDC. VDC, introduced with AuroraCap (Chai et al., 2025), evaluates detailed video captioning using structured reference descriptions. We follow the official VDC QA evaluation pipeline, using Llama-3.1-8B-Instruct for semantic yes/no judgments. We report Acc. and Score, denoted $VDC_{\text{Acc.}}$ and VDC_{Score}, respectively. Detailed captioning requires the model to describe several aspects of the scene within one response and provides a complementary evaluation to video question answering. Subsequent work such as VDC-Agent (Wang et al., 2026) also uses this benchmark to study detailed caption generation.

Average relative performance. Let s_j and $s_j^{\text{Full}} > 0$ denote a method's score and the full model's score on reported metric j. For a table with J task metrics, Avg. is the equal-weighted mean $\text{Avg.} = 100J^{-1}\sum_{j=1}^{J} s_j/s_j^{\text{Full}}$. Every reported task metric contributes once. Thus, $VDC_{\text{Acc.}}$ and VDC_{Score} contribute separately when both appear, and the set of metrics changes when a table reports a smaller benchmark subset. Memory, computation, and latency are excluded from this mean. This normalization makes metrics with different numerical scales comparable without assigning larger weights to benchmarks with more samples.

For context, MMBench and MM-Vet provide complementary multimodal evaluation protocols (Liu et al., 2024d; Yu et al., 2024). We do not substitute these datasets for the official benchmark splits used here.

C Implementation Details

C.1 Reproduction Details of Compared Baselines

We follow the configurations and token-retention settings provided in the public code repositories of the compared methods. All baselines are evaluated using the same benchmark pipeline as FoveaTree, with identical input preprocessing, frame sampling, decoding settings, and hardware.

C.2 Reproduction Details of FoveaTree

Throughout the main paper, per-frame visual grids, attention normalization, and token budgets refer to the backbone's native temporal units: one sampled frame for InternVL3.5 and LLaVA-OneVision, but two sampled frames per unit for Qwen3-VL and Qwen2.5-VL. FoveaTree uses the public checkpoints of InternVL3.5-8B, Qwen3-VL-8B, Qwen2.5-VL-7B, and LLaVA-OneVision-7B without any additional training, calibration, or reference passes. MGDE operates on layer-0 visual features with a root side length of 8 and a merge-gain threshold of 0.005; the per-unit leaf caps follow Eq. (4) with $L_{\max} = 196$ for InternVL3.5 and LLaVA-OneVision, $L_{\max} = H_t W_t$ for Qwen2.5-VL, and $L_{\max} = 120$ for Qwen3-VL. ATP is inserted at decoder layer 18 with the default split score $\kappa_u = \delta_u + \rho_u$ and a maximum quadtree depth of $D = 4$ (Appendix A.3); the same configuration is transferred unchanged to Qwen2.5-VL and LLaVA-OneVision. Inference uses the same batch-one bfloat16, SDPA, and greedy-decoding configuration as the baselines, with `max_new_tokens`=512 for VDC and 8 for all other benchmarks. Efficiency measurements synchronize CUDA timing and exclude warm-up iterations, and all token counts are audited per temporal unit as described in Appendix C.3.

C.3 Token Budget Alignment

Budget definition. Let N_t be the number of original visual tokens of a native temporal unit t, and let $r \in \{50\%, 30\%, 10\%\}$ be the nominal retention ratio. The per-unit budget and the final output count are defined as

$$B_t = \min\big(N_t, \lceil rN_t \rceil\big), \qquad K_t = \min(B_t, M_t), \tag{3}$$

where M_t is the number of tokens surviving MGDE and K_t is the number of tokens finally emitted by ATP. By construction $K_t \le B_t$: the budget is a hard upper bound, not a quota that must be filled. Text tokens are excluded from N_t, and the budget is computed from the original visual grid of each temporal unit, so backbones with dynamic resolution obtain per-unit budgets that follow their actual token counts. Qwen3-VL and Qwen2.5-VL use `temporal_patch_size=2`, so 64 sampled frames form 32 native temporal units per question-video input; the spatial token count of a Qwen2.5-VL unit follows its dynamic resolution. InternVL3.5 and LLaVA-OneVision retain one temporal unit per sampled frame.

Unit-local budget resolution. When $M_t > B_t$, ATP applies a unit-local budget resolver. (i) The frontier is initialized with the unit-level routing node, to which the surviving roots of the restricted MGDE forests are attached. (ii) At each step, the expandable node with the highest split priority κ_u is selected. (iii) If a node has k valid children, a complete expansion increases the frontier size by $k - 1$, and the expansion is applied whenever this increase fits within the remaining budget. (iv) If a complete expansion would cross the budget, let F be the current frontier size and $d =$

$B_t - F$ the remaining slots. ATP replaces the selected node with its d highest-priority children and one *residual cluster* containing all remaining children, yielding $F - 1 + d + 1 = B_t$ frontier entries. (v) Every terminal cluster maps to exactly one output token, so the resolver terminates with exactly B_t outputs. Because quadtree children are spatially disjoint, the terminal clusters form a non-overlapping frontier with unique spatial supports, and restoring temporal-unit and raster order preserves the original video sequence. The residual-cluster mechanism absorbs the granularity of quadtree expansion: exposing a node adds several tokens at once and could otherwise overshoot the budget, whereas a residual cluster never merges subtrees from different parents.

Under-budget behavior. When MGDE leaves fewer tokens than the requested ATP budget, FoveaTree keeps all MGDE survivors and does not synthesize, pad, or borrow tokens: no tokens are borrowed from other temporal units, the MGDE selection itself is not altered, and the final output is $K_t = M_t$. Algorithm 1 summarizes the complete per-unit procedure, and the implementation enforces the contract with a hard check, $\text{count} = \min(B_t, M_t)$, on every temporal unit; the InternVL3.5, Qwen3-VL, Qwen2.5-VL, and LLaVA-OneVision pipelines raise an immediate error instead of silently accepting a violation.

Algorithm 1 Unit-local token budget resolution for one temporal unit.

$N_t \leftarrow$ number of original visual tokens; $B_t \leftarrow \min(N_t, \lceil rN_t \rceil)$
$M_t \leftarrow$ number of tokens after MGDE
if $M_t \le B_t$ **then**
  emit all MGDE survivors and set $K_t \leftarrow M_t$
else
  frontier $\leftarrow$ {unit-level routing node}
  while $|\text{frontier}| < B_t$ **do**
    $u \leftarrow$ expandable node with the highest split priority κ_u
    if a complete expansion of u fits within B_t **then**
      replace u by all of its valid children
    else
      $d \leftarrow B_t - |\text{frontier}|$
      replace u by its d highest-priority children and one residual cluster
      the residual cluster contains all remaining children of u
      break
    end if
  end while
  emit the frontier clusters and set $K_t \leftarrow |\text{frontier}|$
end if
assert $K_t = \min(B_t, M_t)$

MGDE tree capacity. MGDE applies a leaf-count cap to the entire quadtree forest of each temporal unit, rather than separately to each root. Let L_t^{tree} denote its current number of terminal regions. A candidate split is permitted only if the resulting forest satisfies

$$L_t^{\text{tree}} \le L_{\max}. \tag{4}$$

We use $L_{\max} = 196$ for InternVL3.5 and LLaVA-OneVision, $L_{\max} = H_t W_t$ for Qwen2.5-VL, and $L_{\max} = 120$ for Qwen3-VL, where $H_t \times W_t$ is the unit's visual grid. The cap is an upper bound, not a prescribed leaf count: the split threshold and forest structure determine where refinement stops. It is distinct from the ATP output budget B_t. Subsequent density-guided deletion and merging can further reduce the number of survivors M_t, producing the unfilled budgets reported below.

Table 13: Definition of the per-unit token-budget symbols.

Symbol	Meaning
N_t	number of original visual tokens
r	nominal retention ratio
B_t	ATP budget upper bound, $\min(N_t, \lceil rN_t \rceil)$
M_t	number of tokens surviving MGDE
K_t	number of tokens finally emitted by ATP
$B_t - K_t$	budget left unfilled by MGDE
$K_t - B_t$	signed budget excess, must be ≤ 0

Budget auditing. Table 13 summarizes the notation. Table 14 audits the actual token counts over the full Video-MME evaluation for each backbone: InternVL3.5, Qwen3-VL, Qwen2.5-VL, and LLaVA-OneVision. Each evaluation covers 2,700 questions over 900 videos. For InternVL3.5,

the audit contains 172,800 temporal units across 2,700 question-video inputs. On this run, MGDE leaves a mean of 187.38 survivors per 256-token temporal unit, the share of units below budget is 4.45%, 1.93%, and 0.52% at 50%, 30%, and 10% retention, and all 172,800 InternVL3.5 units satisfy $K_t = \min(B_t, M_t)$ with zero over-budget units. Across all audited backbones, no temporal unit ever exceeds its budget; every deviation is an unfilled budget, i.e., a case in which MGDE produces fewer survivors than the budget allows. Each input's total budget is the sum of its per-unit budgets; for instance, 30% retention on LLaVA-OneVision yields $64 \times \lceil 0.30 \times 196 \rceil = 3{,}776$ tokens. Unit-level and input-level statistics are kept separate: the large input-level deficits, such as 3,144 tokens at 50% retention, aggregate the per-unit under-fills across the temporal units in an input and are not deficits of any single unit.

Table 14: Token-budget audit over all 2,700 Video-MME question-video inputs for each backbone. Actual is the mean final visual-token count per input; Range is its minimum and maximum. Unfilled and Max unfilled are the mean and maximum of each input's own budget minus its final token count. Util. is total retained tokens divided by total budget. Units counts native temporal units across all inputs; Over-budget counts units exceeding their own budgets. Budgets are computed separately for each input, including dynamic-resolution inputs.

Model	Ret.	Units	Actual	Range	Unfilled	Max unfilled	Util.	Over-budget
InternVL3.5	50%	172,800	8,043.21	5,048–8,192	148.79	3,144	98.2%	0
InternVL3.5	30%	172,800	4,881.93	3,591–4,928	46.07	1,337	99.1%	0
InternVL3.5	10%	172,800	1,656.70	1,326–1,664	7.30	338	99.6%	0
LLaVA-OneVision	30%	172,800	3,628.93	1,685–3,776	147.07	2,091	96.105%	0
LLaVA-OneVision	10%	172,800	1,251.26	795–1,280	28.74	485	97.755%	0
Qwen3-VL	30%	86,400	1,393.19	956–1,440	12.85	452	99.086%	0
Qwen3-VL	10%	86,400	477.97	375–480	1.31	105	99.726%	0
Qwen2.5-VL	30%	86,400	1,716.51	1,318–1,888	9.46	410	99.452%	0
Qwen2.5-VL	10%	86,400	574.09	450–640	1.34	126	99.767%	0

Fairness of the comparison. All methods are evaluated under the same nominal maximum output-token budget. ToMe, STTM, and FlashVID fill this budget exactly, whereas FoveaTree enforces it as a hard upper bound and retains all MGDE survivors without padding when fewer are available. This is a same-maximum-budget comparison, not a per-sample identical-token-count comparison; Table 14 reports FoveaTree's actual budget utilization. An equal-or-lower final token count does not by itself imply an equal-or-lower total inference cost, because methods differ in their pruning locations and overhead. FLOPs are computed for the actual execution, accounting for the sequence lengths processed at each stage and the pruning operations; latency and peak memory are measured over the full inference pipeline, including tree construction, attention extraction, and pruning.

C.4 Properties of Tree-Based Aggregation

Weighted merging of terminal clusters. For a terminal cluster C, the weights $w_v = (\rho_v + \epsilon)/\sum_{v' \in C}(\rho_{v'} + \epsilon)$ are positive and sum to one. The merged hidden state $\widehat{h}_C^\ell = \sum_{v \in C} w_v h_v^\ell$ lies in the convex hull of its constituent states. The constituent with the highest attention provides the position anchor, and the merged state retains that constituent's original position under RoPE (Su et al., 2024). In particular, $\|\widehat{h}_C^\ell\|_2 \le \max_{v \in C} \|h_v^\ell\|_2$. For a constituent state h_i^ℓ, the triangle inequality also gives $\|h_i^\ell - \widehat{h}_C^\ell\|_2 \le \sum_{v \in C} w_v \|h_i^\ell - h_v^\ell\|_2$. These relations explain why homogeneous groups admit a compact representation and motivate combining heterogeneity with attention when deciding to merge.

Stability of bottom-up propagation. For a fixed surviving tree, consider two sets of normalized leaf scores ρ and ρ'. At any internal node u, maximum propagation satisfies

$$\left| \max_{v \in \mathcal{V}(u)} \rho_v - \max_{v \in \mathcal{V}(u)} \rho'_v \right| \le \max_{v \in \mathcal{V}(u)} |\rho_v - \rho'_v|. \tag{5}$$

To see this, let v^* maximize the first score set. The first maximum minus the second is at most $\rho_{v^*} - \rho'_{v^*}$. Exchanging the two score sets gives the reverse bound. Propagating a maximum therefore does not amplify the largest perturbation of the normalized leaf scores on a fixed tree.

Moreover, $\rho_u \geq \rho_v$ for every surviving descendant v. A region containing strong evidence cannot receive a smaller propagated relevance merely because it also contains weakly attended units. This property explains the preservation of regional support under maximum aggregation. It applies after the within-unit percentile normalization and does not establish a corresponding bound on the raw attention distribution or on the final answer accuracy.

D Related Work

Visual representations and video LLMs. Vision Transformers (Dosovitskiy et al., 2021) and hierarchical variants such as Swin (Liu et al., 2021) encode images as patch sequences; CLIP and SigLIP (Radford et al., 2021; Zhai et al., 2023) use language supervision; MAE, DINOv2, and DINO (He et al., 2022; Oquab et al., 2024; Caron et al., 2021) use self-supervision; and BLIP, BLIP-2, Flamingo, and LLaVA (Li et al., 2022; 2023; Alayrac et al., 2022; Liu et al., 2023; 2024a) connect visual encoders to language models. Video-LLaVA, Video-ChatGPT, Chat-UniVi, TimeChat, LLaMA-VID, and InternVideo2 (Lin et al., 2024; Maaz et al., 2024; Jin et al., 2024; Ren et al., 2024; Li et al., 2024b; Wang et al., 2024b) extend multimodal modeling to video; Qwen-VL, Qwen2-VL, and LLaVA-UHD (Bai et al., 2023; Wang et al., 2024a; Xu et al., 2024) accommodate varying image resolutions and aspect ratios; and earlier InternVL releases and ShareGPT4V (Chen et al., 2024d;c;b) add open-source resources for visual instruction tuning. We evaluate InternVL3.5-8B, Qwen3-VL-8B, Qwen2.5-VL-7B, and LLaVA-OneVision-7B (Wang et al., 2025; Bai et al., 2025a;b; Li et al., 2025).

Token selection and compression. TokenLearner learns compact visual tokens (Ryoo et al., 2021), DynamicViT and ATS adaptively select tokens (Rao et al., 2021; Fayyaz et al., 2022), and EViT and Token Pooling consolidate redundant information (Liang et al., 2022; Marin et al., 2023). In VLMs, LLaVA-PruMerge combines selection and merging (Shang et al., 2025), HiRED allocates image-partition budgets (Arif et al., 2025), and VisPruner balances saliency and diversity (Zhang et al., 2025a); ZipVL performs dynamic token sparsification (He et al., 2024), and FocusLLaVA and DeCo study coarse-to-fine or decoupled compression (Zhu et al., 2024; Yao et al., 2024). PyramidDrop, SparseVLM, and ATP-LLaVA prune across decoder layers (Xing et al., 2025; Zhang et al., 2025b; Ye et al., 2025); MADTP uses multimodal alignment (Cao et al., 2024); and DART, FEATHER, HoloV, and MustDrop target duplicate features, spatial coverage, holistic context, and training-free reduction (Wen et al., 2025; Endo et al., 2025; Zou et al., 2025; Han et al., 2026). Our baselines include ToMe's similarity merging (Bolya et al., 2023), FastV's decoder attention pruning (Chen et al., 2024a), VisionZip's dominant and contextual selection (Yang et al., 2025), FrameFusion's combined similarity and importance (Fu et al., 2025b), STTM's multi-granular spatiotemporal merging (Hyun et al., 2025), FlashVID's attention- and diversity-based tree merging (Fan et al., 2026), and LearnPruner's learned module with decoder attention pruning (Takezoe et al., 2026). Attention sinks (Xiao et al., 2024) and position-sensitive retrieval (Liu et al., 2024b) are documented in LLMs; VLM debiasing methods address biased token importance (Zhao et al., 2025a; Zhang et al., 2026). FoveaTree reuses distortion-based spatial quadtrees for query-guided pruning without training.

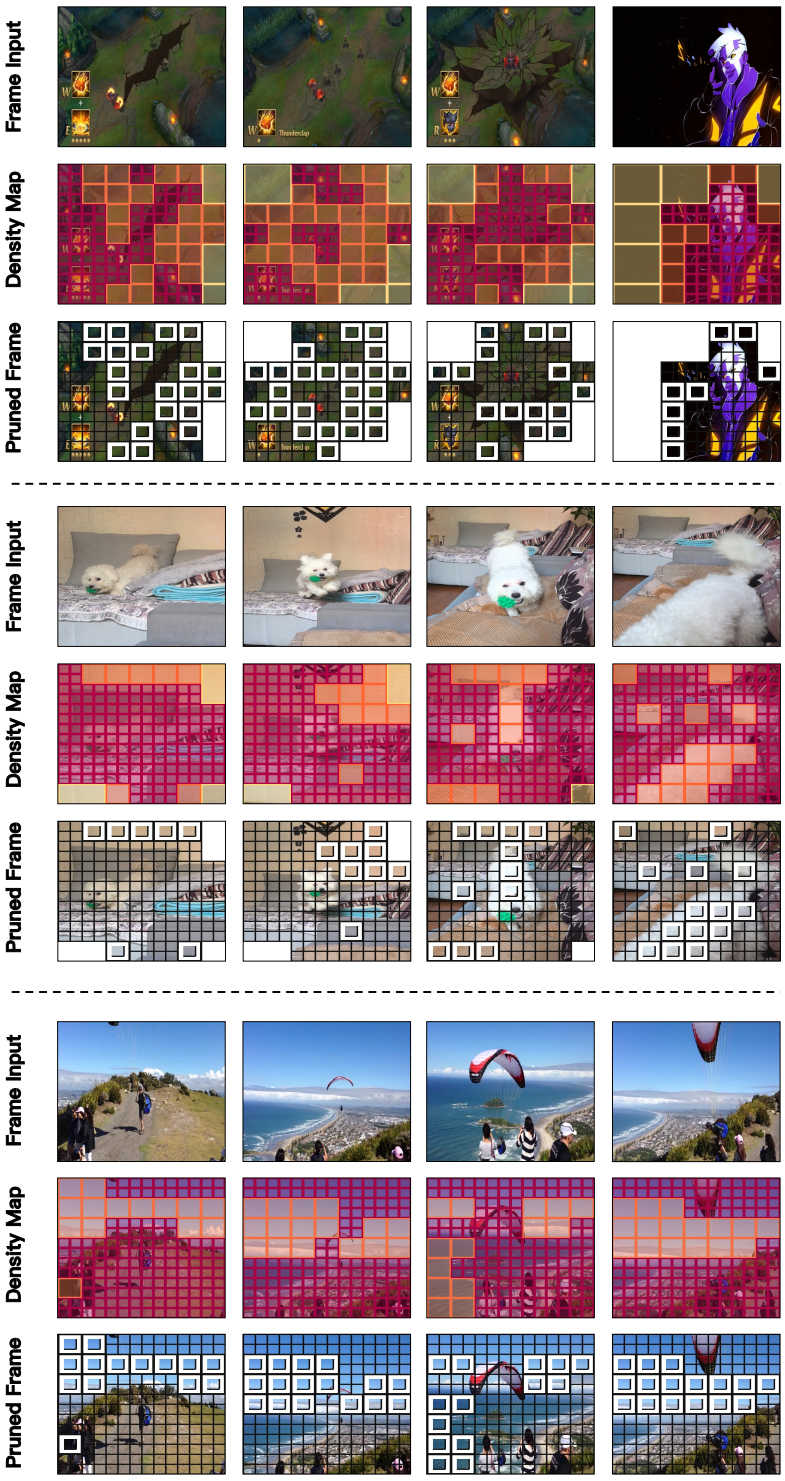
Related analyses cover visual shortcomings and internal processing in multimodal models (Tong et al., 2024; Neo et al., 2025), high-resolution processing (Li et al., 2026), and the Transformer attention underlying decoder-side selection (Vaswani et al., 2017).

E More Visualizations

We visualize the spatial compression performed by MGDE and ATP and compare the responses of InternVL3.5 and Qwen3-VL with and without FoveaTree. In the compressed-frame displays, filled grid cells mark retained tokens, small inset patches represent merged regions, and empty regions indicate removed content.

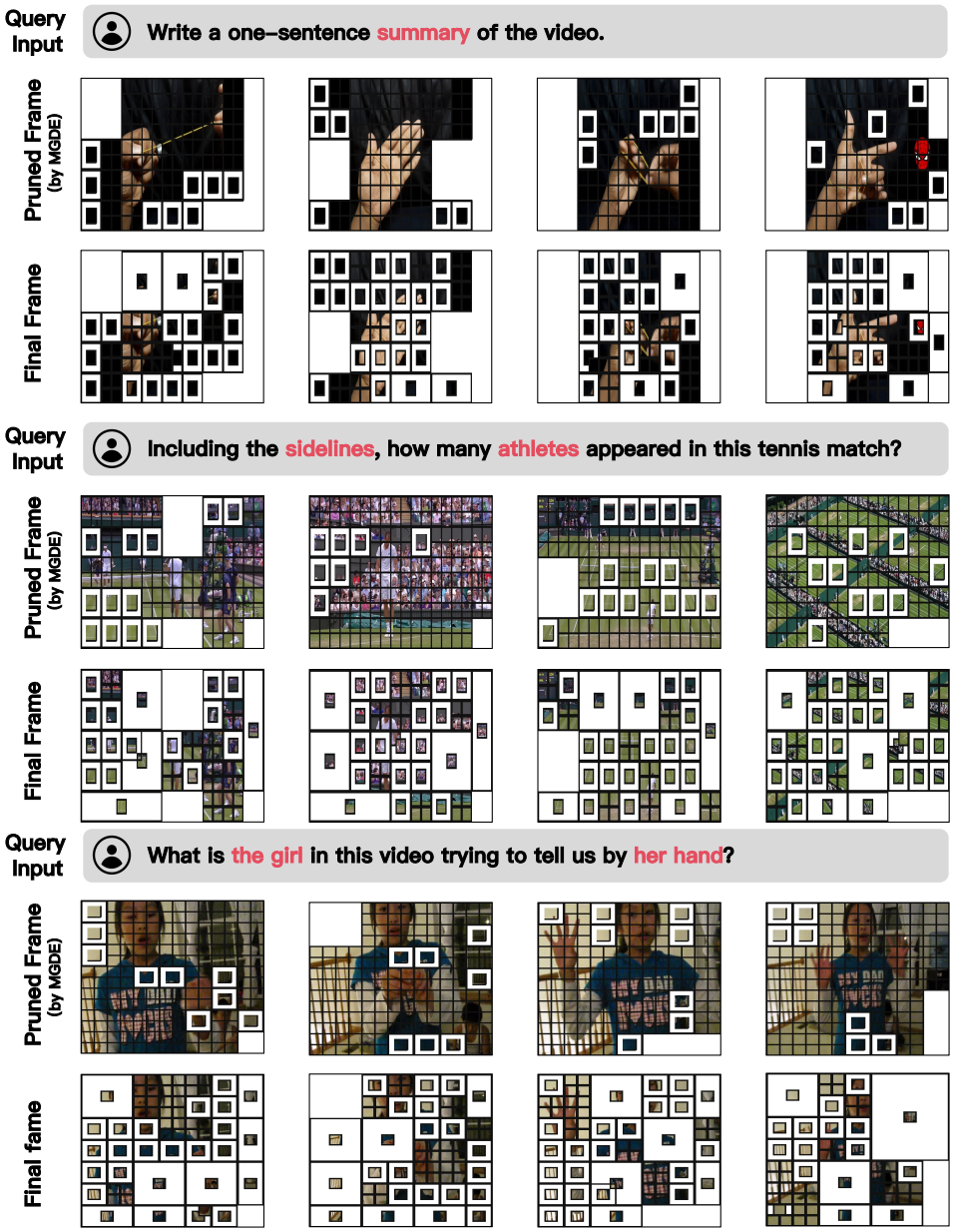
E.1 VISUALIZATIONS OF MGDE

Each figure arranges input frames, MGDE density maps, and compressed representations from top to bottom. Yellow, orange, and magenta mark coarse, medium, and fine terminal regions, i.e., dropped, merged, and retained tokens. MGDE assigns finer partitions to detailed regions and coarser ones to homogeneous regions.



E.2 VISUALIZATIONS OF ATP

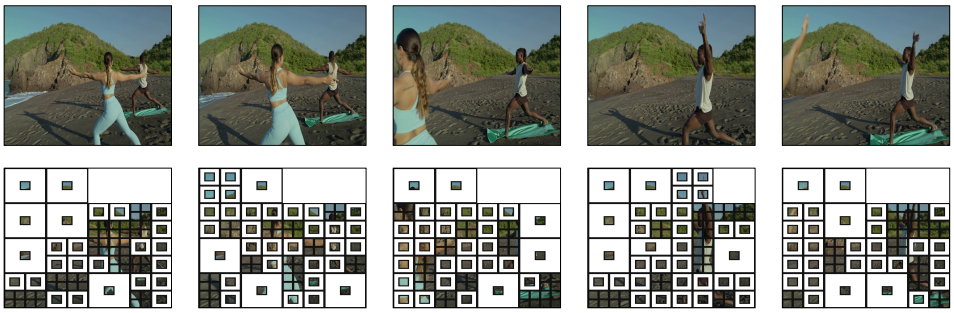
Each example shows the query above four sampled frames, with the MGDE output in the upper row and the final ATP output in the lower row. The queries concern a hand gesture, the number of athletes in a tennis match, and a summary of a manual activity. ATP uses the query and inherited spatial hierarchy to further compress the MGDE survivors. The final representations retain regions around the girl's face and hands, people in the tennis scene, and the hands and objects involved in the demonstrated activity, respectively, illustrating the spatial allocation of the final token budget.



E.3 QUALITATIVE ANALYSIS ON INTERNVL3.5

The two examples compare InternVL3.5-8B with and without FoveaTree, showing the sampled frames, compressed representations, and model responses. In the yoga example, FoveaTree describes two women practicing yoga on the beach, while the full model misidentifies one participant as a man. In the emotion question, the answer changes from "Unable to determine" to "Full of

pressure." The retained regions cover the people and their poses, illustrating how compressed visual inputs can still support detailed descriptions and question answering.



Question: Write a concise one–sentence description that encapsulates the essence of the video.

Intern3.5–VL: Woman y does a yoga on the side stretch on the beach with man. ✗

+FoveaTree: Woman 1 and Woman 2 are doing yoga on the beach. ✓



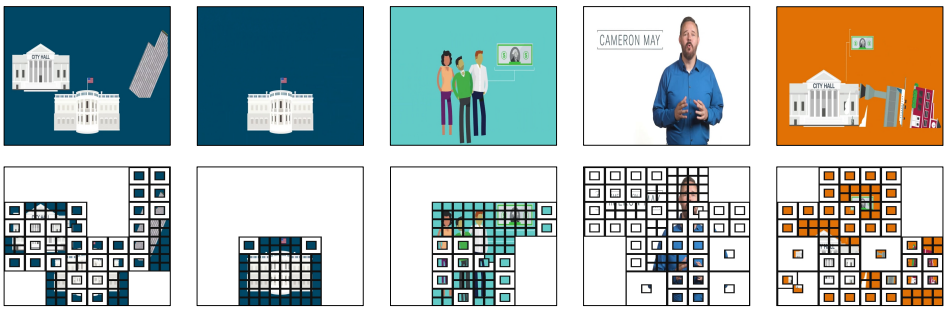
Question: How do the people in the video appear emotionally?

Intern3.5–VL: B. Unable to determine. ✗

+FoveaTree: C. Full of pressure. ✓

E.4 Qualitative Analysis on Qwen3-VL

The Qwen3-VL-8B examples cover numerical question answering and video captioning. For the financial-video question, FoveaTree selects the correct $10 trillion difference between the two markets, while the full model selects $5 trillion. In the balloon example, FoveaTree describes the girl tossing the balloon above her head and catching it, capturing the action sequence more precisely than the full model's description of throwing it away.



Question: Based on data presented in the video, what was the approximate difference in value between the US bond market and the US stock market as of 2020?

Qwen3–VL: A. $5 trillion. ✗

+FoveaTree: B. $10 trillion. ✓



Question: Write a concise one-sentence description that encapsulates the essence of the video.

Qwen3-VL: A little girl is happily playing with a yellow balloon and thorwing it away ✗

+FoveaTree: A little girl tosses a yellow balloon above her head and then catches it in her arms. ✓