GLANCE: EFFICIENT VIDEO LARGE LANGUAGE MODELS VIA SCREENING AND PARALLEL REDUNDANCY SUPPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Denser frame sampling improves temporal coverage in video large language models (Video-LLMs) but makes decoder prefill expensive. Visual-token compression must balance subset quality with selection cost: token-wise ranking can retain redundant evidence, whereas subset-aware selection incurs pairwise comparisons and sequential updates. We introduce GLANCE, a training-free, query-agnostic video token compressor that compares redundancy within compact candidate pools. Informativeness and Uniqueness-guided Screening (IUS) combines within-group feature-response ranks with video-wide directional uniqueness to shortlist candidates. Parallel Redundancy Suppression (PRS) discounts similarity to earlier candidates in a fixed utility order, computing penalties in parallel without iterative selected-set updates. Across three Video-LLM backbones and four benchmarks, the complete configuration, with approximately 12.5% decoder-input retention and late visual-token withdrawal at nominal 10% retention, preserves 94.2–97.6% of full-token mean accuracy with 2.2–3.9$\times$ speedups in model-side time to first token (TTFT). In a matched Qwen3-VL comparison, PRS reduces candidate-selection time from 8.37 to 0.66 ms at 32 frames with close observed accuracy. Compressed 128-frame input also exceeds the 64-frame full-token baseline in three-task mean accuracy at lower TTFT on Qwen3-VL.Code is available at `https://anonymous.4open.science/r/glance-CEE3`

1 INTRODUCTION

Video large language models (Video-LLMs) extend multimodal reasoning from static images to temporally evolving visual content, enabling tasks such as video question answering, event understanding, and long-video reasoning (Zhang et al., 2024b; Bai et al., 2025). Their performance depends on how much of the video is observed. Sampling more frames improves temporal coverage and can capture short or sparse events, but the additional visual tokens increase computation and memory during decoder prefill. In our Qwen3-VL-8B-Instruct evaluation, increasing the input from 64 to 256 frames improves the full-token mean on VideoMME-long, LongVideoBench, and MLVU from 62.4 to 67.4, while model-side time to first token (TTFT) increases from 1,383.0 to 9,730.1 ms (Section 4.4). This tension between temporal coverage and inference cost has motivated training-free visual token compression, which aims to preserve useful video evidence while reducing the number of tokens processed by the decoder.

Existing training-free methods improve video token compression through importance estimation, redundancy reduction, temporal allocation, token merging, or decoder-side pruning (Yang et al., 2025; Shen et al., 2026; Fan et al., 2026; Liu et al., 2025; Lin et al., 2026; 2025). However, nominal retention alone does not determine inference efficiency. Figure 1(b) compares complete inference configurations, whose accuracy–latency trade-offs reflect method-specific token budgets, decoder schedules, and selection overhead. Selection itself contributes to inference cost: attention extraction, pairwise comparison, multi-stage merging, and iterative subset updates all introduce additional computation before the compressed tokens reach the decoder. A useful compression method must therefore preserve visual evidence while keeping the cost of constructing the subset sufficiently small to realize a net latency benefit.

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.

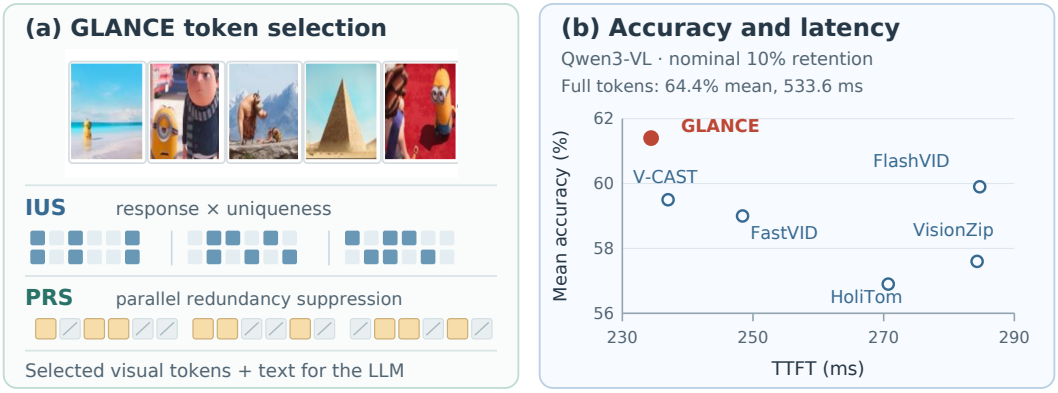


Figure 1: **GLANCE token selection and latency.** **(a)** From sampled video frames, IUS screens frozen visual features using response strength and global uniqueness; PRS suppresses redundancy within each group's shortlist. Token glyphs and counts are schematic. **(b)** Measured four-benchmark mean accuracy and TTFT on Qwen3-VL at nominal 10% retention (Table 1). Points represent complete inference configurations; the upper-left region is preferred. Axes are truncated, and the full-token reference is annotated above the plot.

This creates a tension in video token selection. Token-wise ranking is computationally attractive because it assigns scores without updating them against the selected subset, but a high individual score does not guarantee high marginal value in the final subset. Several high-scoring tokens may represent nearly identical content and consume the budget redundantly. Subset-aware selection can address this issue by explicitly reasoning about relations among candidates, but doing so over the full token population can require quadratic pairwise comparisons or sequential updates as the selected set grows (Alvar et al., 2025; Tan et al., 2026; Zhang et al., 2026). The key question is therefore: *how can we obtain an informative and complementary visual evidence subset without performing expensive relational reasoning over the full set of video tokens?* We argue that relational selection should be concentrated on a small set of plausible candidates rather than applied uniformly to all input tokens.

To address this challenge, we introduce **GLANCE**, a training-free and query-agnostic video token compressor built around a simple principle: *screen broadly, compare selectively*. As illustrated in Figure 1(a), Informativeness and Uniqueness-guided Screening (IUS) combines within-group feature-response ranks with video-wide directional uniqueness to shortlist promising candidates. Uniform group quotas distribute the token budget across the sampled video, while the scores determine which candidates enter the subsequent refinement. This screening reduces the number of tokens requiring explicit comparison, leaving candidate-level redundancy suppression to the second stage.

The second stage resolves redundancy only among these shortlisted candidates. GLANCE introduces Parallel Redundancy Suppression (PRS), which first fixes the candidate order according to the screening scores and then discounts candidates that are highly similar to earlier candidates in that fixed order. Because the reference order is fixed, all redundancy penalties can be computed with batched matrix operations, avoiding the iterative selected-set updates required by greedy subset construction. This design encourages complementary selection while confining pairwise comparison to a small, budget-controlled candidate space. GLANCE thus uses inexpensive scoring across all tokens and reserves explicit redundancy comparisons for promising candidates.

We evaluate GLANCE across three Video-LLM backbones and four video understanding benchmarks. At nominal 10% retention, the complete configuration, with approximately 12.5% decoder-input retention and late visual-token withdrawal, preserves 94.2–97.6% of the full-token mean accuracy while achieving 2.2–3.9× speedups in model-side TTFT. The resulting accuracy–latency trade-off also supports denser temporal observation: on Qwen3-VL, GLANCE with 128 input frames reaches a three-task mean of 63.0 at 981.3 ms TTFT, exceeding the 62.4 mean of the 64-frame full-token model while remaining faster than its 1,383.0 ms TTFT. A matched comparison with greedy subset construction further shows that PRS reduces candidate-selection time from 8.37 to 0.66 ms at 32

frames with close observed accuracy (Section 4.6). These results support screening before redundancy suppression to balance subset quality and construction cost.

2 Related Work

Video models and compact representations. Video-LLMs extend visual instruction following to temporal inputs (Lin et al., 2024; Li et al., 2024a; Zhang et al., 2024b; Bai et al., 2025). Learned interfaces compress visual representations through querying or fusion (Li et al., 2023; 2024c; Zhang et al., 2025c); token sparsification and merging reduce existing features (Rao et al., 2021; Bolya et al., 2022; Cao et al., 2023).

Visual evidence and temporal redundancy. Visual token reduction uses encoder saliency, similarity, and contextual cues (Shang et al., 2025; Yang et al., 2025; Zhang et al., 2025b; Yu et al., 2026). For video, segmentation, spatiotemporal merging, and context aggregation exploit redundancy across frames (Tao et al., 2025; Hyun et al., 2025; Fu et al., 2025b; Shao et al., 2026; Li et al., 2026b). FastVID dynamically partitions videos into temporal segments, then combines density-based token merging with attention-based token selection within each segment (Shen et al., 2026). FlashVID combines frame-wise attention- and diversity-based selection with tree-based spatiotemporal merging (Fan et al., 2026). Budget allocation also matters: VidCom^2 sets frame budgets from video-level uniqueness, MMG-Vid optimizes segment-level marginal gains, and V-CAST allocates tokens using semantic trajectory curvature (Liu et al., 2025; Ma et al., 2026; Lin et al., 2026). Query-conditioned methods additionally use task information (Huang et al., 2025; Li et al., 2026c;a). VidCom^2 sums negative cosine similarities to the frame and video means, formed from the original features (Liu et al., 2025). V-CAST adds cosine distance to the frame mean to a min–max-normalized feature norm (Lin et al., 2026).

Complementary subset selection. Diversity-aware selection connects to farthest-first clustering and maximal marginal relevance (MMR), which update coverage or redundancy as the selected set grows (Gonzalez, 1985; Carbonell & Goldstein, 1998). For visual tokens, DivPrune uses greedy max–min diversity, IDPruner balances importance and redundancy through MMR, and CDPruner uses greedy conditional determinantal selection (Alvar et al., 2025; Tan et al., 2026; Zhang et al., 2026). These methods update scores against the selected subset and thus select tokens sequentially. Detection suppression decays overlapping-box scores sequentially in Soft-NMS (Bodla et al., 2017) or in parallel in Matrix NMS (Wang et al., 2020).

Visual computation across decoder depth. FastV, SparseVLM, and PyramidDrop reduce visual tokens inside the decoder (Chen et al., 2024; Zhang et al., 2024a; Xing et al., 2024); AdaTP, FitPrune, and VScan refine attention signals, pruning schedules, or compression across stages (Sun et al., 2025; Ye et al., 2025; Zhang et al., 2025a). VTW withdraws visual tokens after information migration, while ShortV freezes their updates in layers with low contribution (Lin et al., 2025; Yuan et al., 2025). TwigVLM and SpecVLM also connect pruning to speculative decoding (Shao et al., 2025; Ji et al., 2025).

3 Method

GLANCE uses IUS to screen frozen visual features into per-group shortlists and PRS to suppress redundancy within them in parallel (Figures 1(a) and 2). Both stages are independent of the question and decoder attention maps.

The backbone maps T sampled decoded frames to F visual groups, each potentially combining multiple frames. Let $\mathbf{v}_{f,p} \in \mathbb{R}^D$ denote the original decoder-ready feature of token p in group f, with P tokens per group and $N = FP$ tokens in total. We allocate the input budget B uniformly across groups: their quotas k_f sum to B and differ by at most one token. Quotas distribute the budget across temporal groups; scores determine the candidates within each group.

3.1 Informativeness and Uniqueness-guided Screening

IUS combines within-group response strength and video-wide uniqueness without a video-wide pairwise matrix (Figure 2(A–B)).

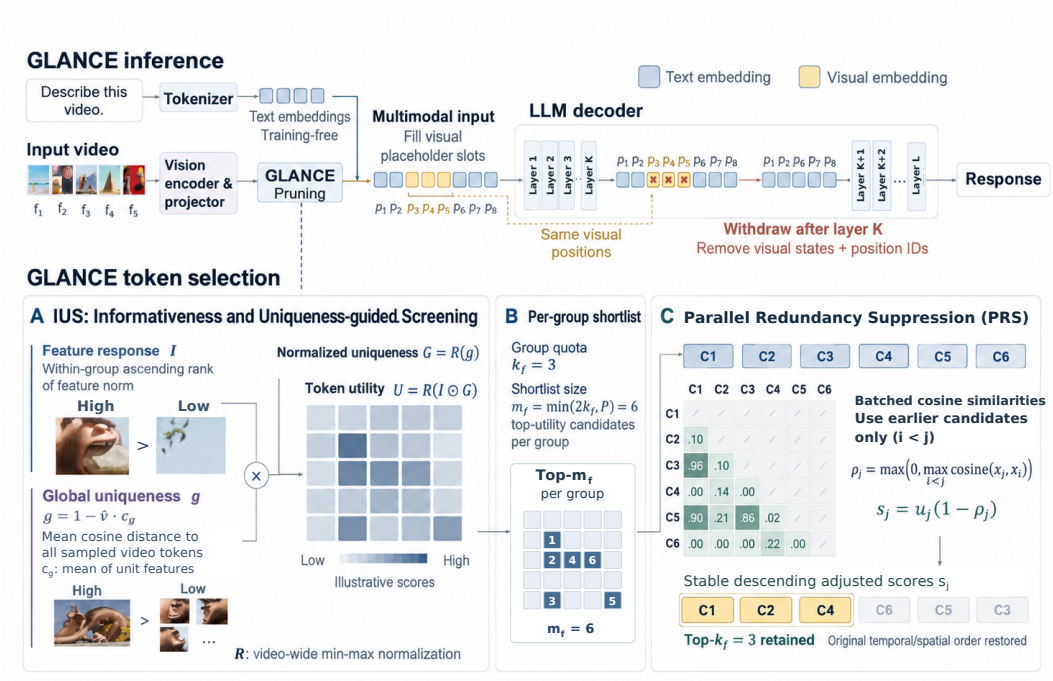


Figure 2: **GLANCE inference and token selection.** Above, selected features and original position indices join text embeddings before prefill; visual states and positions are withdrawn after layer K. **(A)** IUS combines within-group feature-norm rank I with mean cosine distance to all video tokens g, computed from unit-normalized features. Min–max normalization gives $G = \mathcal{R}(g)$ and $U = \mathcal{R}(I \odot G)$. **(B)** For $k_f = 3$, the six darkest cells of the 5×5 utility map in (A) form the shortlist at the same grid positions. Numbers 1–6 correspond to C1–C6 in (C). **(C)** PRS discounts similarity to earlier candidates in fixed utility order, then selects by adjusted score. C3's similarity to C1 lowers its score below C4, retaining C1, C2, and C4. Crops and scores are schematic.

Response strength. We measure response strength by the Euclidean norm $\|\mathbf{v}_{f,p}\|_2$ of the original feature vector, before feature normalization. We rank these magnitudes within each group to limit the effect of scale differences between groups. Let $a_{f,p} \in \{0, \ldots, P-1\}$ be this ascending, zero-based rank: the smallest norm receives rank zero and the largest receives rank $P-1$. The normalized rank is

$$I_{f,p} = \frac{a_{f,p}}{P-1}. \tag{1}$$

Thus $I_{f,p}$ lies in $[0, 1]$ and serves as a relative feature-response proxy for screening.

Global uniqueness. To downweight repeated directions, we measure global uniqueness. Let $\hat{\mathbf{v}}_{f,p} = \mathbf{v}_{f,p}/\|\mathbf{v}_{f,p}\|_2$ be the unit-normalized feature, and let $\mathbf{c}_g = N^{-1} \sum_{f',p'} \hat{\mathbf{v}}_{f',p'}$ be the mean over all N sampled video tokens. The raw uniqueness score is

$$g_{f,p} = 1 - \hat{\mathbf{v}}_{f,p}^{\top} \mathbf{c}_g = \frac{1}{N} \sum_{f',p'} \left(1 - \hat{\mathbf{v}}_{f,p}^{\top} \hat{\mathbf{v}}_{f',p'}\right). \tag{2}$$

Because $\mathbf{c}_g$ is not renormalized, $g_{f,p}$ is the mean cosine distance to all sampled tokens, including the token itself: a larger value indicates lower average directional alignment with the sampled video. This coarse video-wide cue costs $\mathcal{O}(ND)$; PRS handles explicit candidate comparisons. VidCom^2 instead computes cosine similarity to a mean of the original features (Liu et al., 2025); the normalization order makes the two scores different.

Candidate screening. Global uniqueness discounts common feature directions, while response ranks downweight distinctive tokens with weak responses. Their product favors candidates supported by both cues. Let $\mathcal{R}$ denote min–max normalization of a token-score vector $z \in \mathbb{R}^N$ over *all* video

tokens:

$$\mathcal{R}(z)_{f,p} = \frac{z_{f,p} - \min_{f',p'} z_{f',p'}}{\max_{f',p'} z_{f',p'} - \min_{f',p'} z_{f',p'}}. \tag{3}$$

We normalize the uniqueness scores to obtain G, then normalize their elementwise product with I to obtain utility U:

$$G = \mathcal{R}(g), \qquad U = \mathcal{R}(I \odot G), \tag{4}$$

where $I, G, U \in \mathbb{R}^N$ and $\odot$ denotes elementwise multiplication. For group f, IUS retains the $m_f = \min(2k_f, P)$ highest-utility candidates. The extra candidates give PRS alternatives to similar tokens without revisiting the full population. Appendix C.4 details quota allocation and the selection implementation.

3.2 Parallel Redundancy Suppression

High individual utility does not ensure complementary evidence in the retained subset. PRS therefore compares candidates within each shortlist and fixes their reference order so that all penalties can be computed in parallel.

Parallel discounts. Within each group, `topk` returns candidates in nonincreasing utility order, which remains fixed throughout suppression. Let $\mathbf{x}_j$ and u_j denote the unit-normalized feature and utility of the jth candidate in this order. We define its redundancy penalty and adjusted utility as

$$\rho_j = \begin{cases} 0, & j = 1, \\ \max\{0, \max_{i<j} \mathbf{x}_j^\top \mathbf{x}_i\}, & j > 1, \end{cases} \qquad s_j = u_j(1 - \rho_j). \tag{5}$$

The largest positive cosine similarity to an earlier candidate determines the discount.

We form candidate similarity matrices by batched multiplication, mask the diagonal and upper triangle, and take row maxima clamped at zero (Figure 2(C)). All penalties are computed together without iterative selected-set updates.

Subset construction. We retain the k_f candidates with the highest adjusted utilities, breaking ties by the initial candidate order. The reference set includes all earlier candidates, even those ultimately discarded, which removes dependence on the evolving selected subset. Section 4.6 compares the resulting quality–cost trade-off with a matched greedy variant. Pairwise suppression is confined to each group, while IUS supplies the video-wide distinctiveness cue. Finally, we restore the retained tokens to their original temporal and spatial order and pass their original features and position indices to the LLM.

Algorithm 1 in Appendix C.4 summarizes the construction. Including within-group sorting, the selector costs $\mathcal{O}(ND + FP \log P + \sum_f m_f^2 D)$. Only candidate pools incur quadratic similarity computation; groups are chunked to bound temporary memory. At fixed P and retention fraction, cost is linear in F. With input expansion at $r = 10\%$, $k_f \approx P/8$ and $m_f \approx P/4$, so $m_f^2/P^2 \approx 1/16$ (ignoring rounding). This compares similarity-matrix sizes, not measured speedups.

3.3 Inference Configuration

IUS–PRS defines the selector. The complete configuration evaluated in Table 1 additionally uses input-budget expansion and late visual-token withdrawal, following prior pruning and withdrawal strategies (Lin et al., 2025; Fan et al., 2026). For the reported nominal retention settings r, we set $B = \lceil 1.25rN \rceil$ and withdraw visual tokens after $K = \lfloor 0.75L \rfloor$ of the L decoder layers. Text, timestamps, and structural positions continue through the remaining layers according to the backbone's packing convention. At $r = 10\%$, the input fraction is approximately 12.5% and the layer-averaged visual-token fraction is 9.4%. At $r = 20\%$, these fractions are 25% and 18.8%. Section 4.3 separates the selector and schedule contributions. Exact budgets and position handling appear in Appendix C.

4 Experiments

We evaluate GLANCE against existing compression methods across backbones and frame counts, then examine its scoring signals, subset-selection rule, and construction cost.

Table 1: Accuracy at nominal retention $r = 10\%$ and 20% (LLaVA-Video: 10%). Accuracy and Rel. are percentages; TTFT is in ms. Mean averages the four benchmarks and Rel. is relative to full tokens. Bold/underline mark the best/second-best compressed results within each backbone and retention setting. Appendix A gives full retention sweeps; Appendix C.5 details schedules.

Method	Retention r	Benchmark accuracy (%)				Average		TTFT (ms)
		VMME	MVB	LVB	MLVU	Mean	Rel.	
Qwen3-VL-8B-Instruct (32 frames)								
Full tokens	100%	64.4	68.8	60.8	63.7	64.4	100.0	533.6
VisionZip	20%	60.2	63.4	56.9	60.3	60.2	93.5	331.3
FastVID	20%	60.3	65.9	**58.6**	60.1	61.2	95.0	273.5
FlashVID	20%	62.2	66.5	58.0	60.5	61.8	95.9	326.3
HoliTom	20%	58.9	62.0	56.6	60.8	59.6	92.4	300.7
V-CAST	20%	62.7	66.6	58.2	61.7	62.3	96.7	263.5
GLANCE	20%	**63.1**	**68.0**	**58.6**	**62.1**	**63.0**	**97.7**	**261.2**
VisionZip	10%	57.9	59.5	55.6	57.3	57.6	89.4	284.3
FastVID	10%	57.2	63.5	56.8	58.3	59.0	91.5	248.4
FlashVID	10%	59.6	**65.4**	55.9	58.8	59.9	93.0	284.7
HoliTom	10%	56.7	58.3	54.6	57.9	56.9	88.3	270.7
V-CAST	10%	59.6	63.2	56.3	58.8	59.5	92.3	237.0
GLANCE	10%	**61.9**	65.2	**57.6**	**61.1**	**61.4**	**95.3**	**234.4**
Qwen3.8-27B (32 frames)								
Full tokens	100%	68.7	71.2	63.6	65.1	67.2	100.0	2635.4
VisionZip	20%	65.6	67.9	62.1	**63.7**	64.8	96.5	911.4
FastVID	20%	62.3	66.8	59.5	61.6	62.6	93.1	842.3
VidCom^2	20%	64.7	67.9	59.2	62.1	63.5	94.5	838.2
V-CAST	20%	62.9	68.1	61.8	63.2	64.0	95.3	842.5
GLANCE	20%	**65.9**	**69.5**	**63.0**	63.5	**65.5**	**97.5**	**818.8**
VisionZip	10%	61.8	61.4	59.3	61.2	60.9	90.7	737.0
FastVID	10%	59.9	65.8	57.4	61.0	61.0	90.9	698.3
VidCom^2	10%	61.7	63.4	56.7	58.1	60.0	89.3	**673.1**
V-CAST	10%	60.7	65.6	60.3	**62.3**	62.2	92.6	682.1
GLANCE	10%	**63.7**	**66.6**	**61.1**	61.8	**63.3**	**94.2**	676.9
LLaVA-Video-7B (64 frames)								
Full tokens	100%	63.8	61.0	59.2	66.2	62.6	100.0	760.1
VisionZip	10%	58.1	58.5	56.6	61.9	58.8	94.0	382.7
FastVID	10%	59.5	60.0	55.9	62.7	59.5	95.1	397.8
FlashVID	10%	61.0	59.4	57.4	64.6	60.6	96.9	411.3
HoliTom	10%	58.8	59.4	57.3	64.0	59.9	95.7	449.3
V-CAST	10%	56.9	55.0	52.6	58.5	55.8	89.1	364.5
GLANCE	10%	**61.4**	**60.1**	**57.8**	**64.9**	**61.0**	**97.6**	**350.2**

4.1 Experimental Setup

We evaluate Video-MME (Fu et al., 2025a), MVBench (Li et al., 2024b), LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2025) on Qwen3-VL-8B-Instruct (Bai et al., 2025), LLaVA-Video-7B (Zhang et al., 2024b), and Qwen3.8-27B (Qwen Team, 2026). The backbones use 32, 64, and 32 sampled frames, respectively. Table 1 compares GLANCE with full-token inference and locally evaluated training-free baselines. Within each backbone, methods share frame sampling, image processing, prompts, and greedy decoding on NVIDIA H100 80GB GPUs. Appendix C specifies benchmark splits, baseline coverage, and the evaluation protocol.

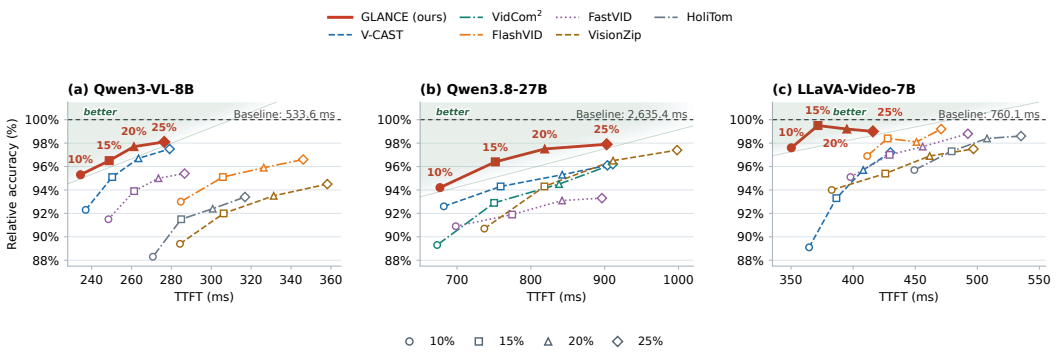


Figure 3: Accuracy–TTFT trade-offs on (a) Qwen3-VL-8B, (b) Qwen3.8-27B, and (c) LLaVA-Video-7B. Higher relative four-benchmark accuracy and lower TTFT are preferred. Markers denote nominal retention of 10%, 15%, 20%, and 25%. Dashed lines and annotations indicate full-token accuracy and TTFT. Vertical axes are truncated; horizontal scales differ.

We report benchmark accuracies, their equal-weight mean, and relative accuracy (Rel.) to the full-token mean at nominal retention settings $r = 10\%$ and 20%; Appendix A includes 15% and 25%. Because each method retains its evaluated inference schedule, nominal r does not imply matched token counts or FLOPs (Appendix C.5). We compare efficiency through measured time to first token (TTFT), including visual encoding, compression, and the first decoder forward pass from prepared inputs; Appendix C.3 defines the timing boundary and aggregation across workers and runs.

4.2 Main Results

The complete GLANCE configuration achieves the highest reported four-benchmark mean among the evaluated compression methods on every backbone at both nominal retention settings (Table 1; LLaVA-Video at 20% in Appendix A.2, Table 6). At $r = 10\%$, its mean exceeds the strongest compressed baseline by 1.5 points on Qwen3-VL, 0.4 on LLaVA-Video, and 1.1 on Qwen3.8-27B. It retains 95.3%, 97.6%, and 94.2% of the respective full-token means, with TTFT speedups of approximately $2.28\times$, $2.17\times$, and $3.89\times$. Its TTFT is comparable to V-CAST on Qwen3-VL (234.4 versus 237.0 ms) and VidCom^2 on Qwen3.8-27B (676.9 versus 673.1 ms).

At $r = 20\%$, GLANCE has the lowest measured TTFT among the compression methods on all three backbones. It retains 97.7%, 99.2%, and 97.5% of the full-token means on Qwen3-VL, LLaVA-Video, and Qwen3.8-27B, respectively. Figure 3 compares complete accuracy–TTFT trade-offs across nominal retention settings, accounting for method-specific budgets and depth schedules through measured latency. Appendix A gives full results and VideoMME duration breakdowns. FlashVID leads at 25% on LLaVA-Video, with a mean of 62.1 versus 61.9.

Main comparisons report point estimates; paired uncertainty analysis covers separate component-ablation cohorts (Appendix D).

4.3 Selector and Inference Schedule

Table 2 separates IUS–PRS from its inference schedule on Qwen3-VL (Appendix B.3). At $r = 10\%$, the selector alone scores 61.1, versus 61.4 for the complete configuration. At $r = 20\%$, it scores 62.3, matching V-CAST's displayed mean, versus 63.0 for the complete configuration. With expanded subsets fixed, withdrawal saves 8.4 and 12.0 ms at 10% and 20%, respectively; means change from 61.4 to 61.4 and from 62.9 to 63.0. The schedule thus contributes to the reported gains; baselines have not been rerun under the same schedule.

4.4 Scaling to More Input Frames

At $r = 10\%$, we scale Qwen3-VL from 64 to 128 and 256 frames on VideoMME-long, LongVideoBench, and MLVU while keeping processor and selection settings fixed. GLANCE's

Table 2: Selector and schedule on Qwen3-VL. All rows use IUS–PRS. B/N is the decoder input fraction, K/L the fraction of layers processing visual tokens, and $\bar{r}$ their product. Mean averages four benchmarks. Values are from Table 14; TTFT differences are single-run measurements.

Configuration	r	B/N	K/L	$\bar{r}$	Mean	TTFT (ms)
Selector only	10%	10%	1	10%	61.1	233.1
Complete configuration	10%	12.5%	3/4	9.4%	61.4	234.4
Selector only	20%	20%	1	20%	62.3	259.2
Complete configuration	20%	25%	3/4	18.8%	63.0	261.2

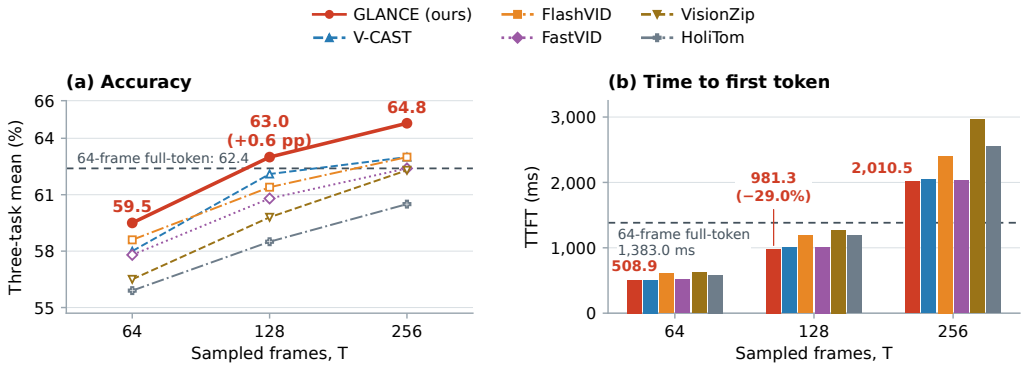


Figure 4: Qwen3-VL frame scaling at $r = 10\%$: (a) mean accuracy on VideoMME-long, LongVideoBench, and MLVU; (b) model-side TTFT. Dashed lines mark the 64-frame full-token reference; annotations compare 128-frame GLANCE with this reference. Values are from Table 10.

three-task mean rises from 59.5 to 63.0 and 64.8, highest among compressed methods at each frame count (Figure 4). At 256 frames, it retains 96.1% of the full-token mean and cuts TTFT from 9,730.1 to 2,010.5 ms (4.8×). Figure 4 highlights compressed 128-frame input exceeding the 64-frame full-token mean (63.0 versus 62.4) at lower TTFT (981.3 versus 1,383.0 ms), with peak memory rising from 23.6 to 27.4 GiB. The fixed retention fraction makes retained token counts grow with frame count.

4.5 Ablation Studies

Complementary scoring signals. We isolate the scoring cues and PRS while fixing the input budget and processing depth within each backbone (Table 3). Combining the two cues improves the four-benchmark mean over norm rank with PRS by 0.4 points on Qwen3-VL and 0.7 on LLaVA-Video, while uniqueness alone falls below random selection on both backbones. This supports combining response ranks, which downweight weak-response outliers, with uniqueness, which discounts common directions; token-level evidence retention remains unmeasured. Removing PRS lowers the means from 61.4 to 60.3 and from 61.0 to 60.6, respectively. Appendices B.1 and E provide task scores and visualizations; Appendix D gives paired accuracy diagnostics on separate cohorts. The gains vary across tasks: on Qwen3-VL, IUS with PRS improves over norm rank with PRS by 2.6 points on VideoMME and 0.8 on MLVU, while MVBench and LongVideoBench decrease by 0.5 and 1.2 points. On LLaVA-Video, three of four tasks improve. Thus the higher mean reflects an aggregate benefit rather than uniform task-level gains.

Candidate-pool size. Table 3(d) shows that expanding the pool from k_f to $2k_f$ improves mean accuracy from 60.3 to 61.4; larger pools increase TTFT without improving the displayed mean. Appendix B.2 reports per-benchmark scores.

Table 3: **Scoring and selection ablations** at $r = 10\%$, with full-token references. Compression variants share budgets and depth. (a,b) Four-benchmark Mean and Rel. to full tokens (%). Single-signal variants retain PRS. (c) Three-task Mean averages displayed full-benchmark scores; timing uses a separate 30-video H100 cohort. (d) Candidate pools with fixed quotas, scoring, and depth. Times are in ms; Select./Compr.: candidate selection/complete compression.

(a) Qwen3-VL, 32 frames

Selection	Mean	Rel.	TTFT
Full tokens	64.4	100.0	533.6
Random	59.5	92.4	234.0
Uniform	59.2	91.9	**233.6**
Uniqueness + PRS	58.5	90.8	234.8
Norm rank + PRS	61.0	94.7	234.8
IUS without PRS	60.3	93.6	234.3
GLANCE	**61.4**	**95.3**	234.4

(b) LLaVA-Video, 64 frames

Selection	Mean	Rel.	TTFT
Full tokens	62.6	100.0	760.1
Random	59.7	95.4	**348.9**
Uniform	59.2	94.6	349.1
Uniqueness + PRS	56.6	90.4	350.1
Norm rank + PRS	60.3	96.3	350.9
IUS without PRS	60.6	96.8	349.4
GLANCE	**61.0**	**97.6**	350.2

(c) Matched selectors: Qwen3-VL, 32 frames

IUS +	VMME	LVB	MLVU	Mean	Select.	Compr.	TTFT
Top-k	60.5	57.1	60.4	59.3	**0.15**	**1.17**	289.85
Greedy	61.7	57.4	60.9	60.0	8.37	9.32	299.77
PRS	**61.9**	**57.6**	**61.1**	**60.2**	0.66	1.68	**288.75**

(d) Candidate pools: Qwen3-VL, 32 frames

Candidates	k_f	$2k_f$ (default)	$4k_f$	All
Mean (%)	60.3	**61.4**	**61.4**	**61.4**
TTFT (ms)	234.6	**234.4**	240.8	244.9

4.6 Efficiency Analysis

With IUS scores, quotas, and schedule fixed, Greedy and PRS compare the same top-$2k_f$ pools (Table 3(c)). Greedy updates its selected set sequentially; PRS uses a fixed order.

PRS reaches 60.2 mean versus 59.3 for Top-k and 60.0 for Greedy, improving every task over Top-k; paired diagnostics do not establish superiority over Greedy (Appendix D). Despite 0.51 ms more compression work, its paired TTFT difference from Top-k is -1.10 ms (95% CI $[-3.09, 0.87]$). The interval spans zero: variation in other stages can mask Top-k's lower compression cost, so a TTFT advantage is not established. Against Greedy, selection time falls from 8.37 to 0.66 ms and TTFT decreases by 11.02 ms (3.7%). Appendices C.6 and D.2 give 128-frame results and paired statistics.

5 Conclusion

GLANCE combines feature-response rank and global uniqueness with parallel redundancy suppression. Ablations support both stages, and profiling shows lower selection cost than Greedy with close observed accuracy. Input expansion and late withdrawal yield favorable accuracy–TTFT trade-offs across three backbones, allowing denser sampling to exceed a sparser full-token reference at lower TTFT, with higher peak memory.

6 Limitations

GLANCE requires features from all sampled frames to score global uniqueness, so the current method supports offline videos; streaming support is future work. Integration into serving engines such as vLLM and SGLang and measurement of their end-to-end speedups also remain future work.

AI USE STATEMENT

We used ChatGPT and Codex to assist with manuscript drafting, translation, and editing; literature search and reference checks; and the creation and revision of scientific figures. AI tools also assisted with consistency checks between mathematical notation and the implementation. The authors take responsibility for the final manuscript, including all AI-assisted content and scientific claims.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

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*, 2025.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In *2017 IEEE international conference on computer vision (ICCV)*, pp. 5562–5570. IEEE, 2017.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12890–12903, 2023.

Jaime G Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, volume 98, pp. 290941–291025, 1998.

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

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

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. *arXiv preprint arXiv:2602.08024*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22654–22663. IEEE, 2025b.

Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

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

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. *arXiv preprint arXiv:2507.07990*, 2025.

Yicheng Ji, Jun Zhang, Heming Xia, Jinpeng Chen, Lidan Shou, Gang Chen, and Huan Li. Specvlm: Enhancing speculative decoding of video llms via verifier-guided token pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7216–7230, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Jiaqi Li, Shuntian Zheng, Yixian Shen, Jia-Hong Huang, Xiaoman Lu, Minzhe Ni, and Yu Guan. Keeping the evidence chain: Semantic evidence allocation for training-free token pruning in video temporal grounding. In *European Conference on Computer Vision*, pp. 549–567. Springer, 2026a.

Jinlong Li, Liyuan Jiang, Haonan Zhang, and Nicu Sebe. Token reduction via local and global contexts optimization for efficient video large language models. *arXiv preprint arXiv:2603.01400*, 2026b.

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 *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024b.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European conference on computer vision*, pp. 323–340. Springer, 2024c.

Yulin Li, Haokun Gui, Ziyang Fan, Junjie Wang, Bin Kang, Bin Chen, and Zhuotao Tian. Less is more, but where? dynamic token compression via llm-guided keyframe prior. *Advances in Neural Information Processing Systems*, 38:156861–156904, 2026c.

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.

Xinying Lin, Xuyang Liu, Yiyu Wang, Teng Ma, and Wenqi Ren. V-cast: Video curvature-aware spatio-temporal pruning for efficient video large language models. *arXiv preprint arXiv:2603.27650*, 2026.

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

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. Video compression commander: Plug-and-play inference acceleration for video large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1910–1924, 2025.

Junpeng Ma, Qizhe Zhang, Ming Lu, Zhibin Wang, Qiang Zhou, Jun Song, and Shanghang Zhang. Mmg-vid: Maximizing marginal gains at segment-level and token-level for efficient video llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24253–24261, 2026.

Qwen Team. Qwen3.8-27B: Model card and checkpoint, 2026. URL `https://huggingface.co/Qwen/Qwen3.8-27B`.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Advances in Neural Information Processing Systems*, 38:135547–135570, 2026.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20064–20074. IEEE, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Fengyuan Sun, Leqi Shen, Hui Chen, Sicheng Zhao, Jungong Han, and Guiguang Ding. Adatp: Attention-debiased token pruning for video large language models. In *EMNLP (Findings)*, pp. 3273–3286, 2025.

Yifan Tan, Yifu Sun, Shirui Huang, Hong Liu, Guanghua Yu, Jianchen Zhu, and Yangdong Deng. Idpruner: Harmonizing importance and diversity in visual token pruning for mllms. *arXiv preprint arXiv:2602.13315*, 2026.

Chong Tang, Sannara Ek, Dirk Koch, Robert Mullins, Alex Weddell, and Jagmohan Chauhan. Surge: Surprise-guided token reduction for efficient video understanding with vlms. In *International Conference on Learning Representations*, volume 2026, pp. 3849–3868, 2026.

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

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Advances in Neural information processing systems*, 33:17721–17732, 2020.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22128–22136, 2025.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free mllm acceleration. *arXiv preprint arXiv:2601.22674*, 2026.

Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. Shortv: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 329–339. IEEE, 2025.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. Vscan: Rethinking visual token reduction for efficient large vision-language models. *arXiv preprint arXiv:2505.22654*, 2025a.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20857–20867. IEEE, 2025b.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Advances in Neural Information Processing Systems*, 38:25438–25468, 2026.

Shaolei Zhang, Qingkai Fang, Yang Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, volume 2025, pp. 53285–53310, 2025c.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024a.

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

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701. IEEE, 2025.

A Complete Experimental Results

We first report the complete retention sweeps for all three backbones, followed by accuracy–latency comparisons and frame-scaling results. Tables 4–9 include every evaluated retention rate, all four benchmarks, and VideoMME duration breakdowns. The means and relative performance follow Equation (7); evaluation settings and token-budget accounting are given in Appendix C.

A.1 Qwen3-VL-8B

Table 4 reports the 32-frame results at $r \in \{10, 15, 20, 25\}\%$, extending the 10% and 20% settings in Table 1. GLANCE attains the highest four-benchmark mean among the compared compression methods at all four retention rates.

Table 4: Comparison on Qwen3-VL-8B-Instruct (32 frames). Accuracy and relative performance are percentages; TTFT is in milliseconds. Within each retention group, bold and underline mark the best and second-best distinct displayed values among compression methods, including ties.

Method	r	VMME	MVB	LVB	MLVU	Mean	Rel.	TTFT
Full tokens	100%	64.4	68.8	60.8	63.7	64.4	100.0	533.6
VisionZip	25%	60.6	65.5	56.9	60.5	60.9	94.5	358.2
FastVID		60.7	66.6	<u>58.3</u>	60.4	61.5	95.4	286.6
FlashVID		62.2	67.2	58.0	61.4	62.2	96.6	346.1
HoliTom		60.4	63.2	56.5	60.6	60.2	93.4	316.9
V-CAST		<u>63.3</u>	<u>67.7</u>	58.1	**62.2**	<u>62.8</u>	<u>97.5</u>	<u>279.1</u>
GLANCE		**63.8**	**68.1**	**59.0**	<u>62.0</u>	**63.2**	**98.1**	**276.4**
VisionZip	20%	60.2	63.4	56.9	60.3	60.2	93.5	331.3
FastVID		60.3	65.9	**58.6**	60.1	61.2	95.0	273.5
FlashVID		62.2	66.5	58.0	60.5	61.8	95.9	326.3
HoliTom		58.9	62.0	56.6	60.8	59.6	92.4	300.7
V-CAST		<u>62.7</u>	<u>66.6</u>	<u>58.2</u>	<u>61.7</u>	<u>62.3</u>	<u>96.7</u>	<u>263.5</u>
GLANCE		**63.1**	**68.0**	**58.6**	**62.1**	**63.0**	**97.7**	**261.2**
VisionZip	15%	59.6	61.7	55.6	<u>60.4</u>	59.3	92.0	306.2
FastVID		59.0	66.0	<u>57.8</u>	59.1	60.5	93.9	261.3
FlashVID		61.5	<u>66.3</u>	57.3	60.1	<u>61.3</u>	<u>95.1</u>	305.8
HoliTom		58.8	60.9	56.4	59.6	58.9	91.5	284.9
V-CAST		<u>62.2</u>	65.5	57.6	59.9	<u>61.3</u>	<u>95.1</u>	<u>250.3</u>
GLANCE		**62.7**	**66.9**	**57.9**	**61.4**	**62.2**	**96.5**	**248.8**
VisionZip	10%	57.9	59.5	55.6	57.3	57.6	89.4	284.3
FastVID		57.2	63.5	<u>56.8</u>	58.3	59.0	91.5	248.4
FlashVID		<u>59.6</u>	**65.4**	55.9	<u>58.8</u>	<u>59.9</u>	<u>93.0</u>	284.7
HoliTom		56.7	58.3	54.6	57.9	56.9	88.3	270.7
V-CAST		<u>59.6</u>	63.2	56.3	<u>58.8</u>	59.5	92.3	<u>237.0</u>
GLANCE		**61.9**	<u>65.2</u>	**57.6**	**61.1**	**61.4**	**95.3**	**234.4**

Table 5: VideoMME duration breakdown for the configurations in Table 4. Scores are percentages.

Method	r	Short	Medium	Long	All
Full tokens	100%	76.3	60.3	56.7	64.4
VisionZip	25%	70.3	57.9	53.6	60.6
FastVID		70.8	57.4	53.9	60.7
FlashVID		73.3	58.9	54.4	62.2
HoliTom		71.2	55.9	54.1	60.4
V-CAST		<u>74.9</u>	<u>60.0</u>	<u>55.1</u>	<u>63.3</u>
GLANCE		**75.0**	**60.7**	**55.8**	**63.8**
VisionZip	20%	68.6	57.2	54.9	60.2
FastVID		70.4	56.1	54.3	60.3
FlashVID		73.3	<u>59.2</u>	54.1	62.2
HoliTom		68.6	55.7	52.3	58.9
V-CAST		<u>73.9</u>	59.1	<u>55.1</u>	<u>62.7</u>
GLANCE		**74.4**	**59.7**	**55.3**	**63.1**
VisionZip	15%	67.8	56.4	54.6	59.6
FastVID		69.3	54.4	53.3	59.0
FlashVID		72.2	57.6	54.7	61.5
HoliTom		69.4	54.9	52.1	58.8
V-CAST		<u>72.6</u>	<u>58.4</u>	**55.7**	<u>62.2</u>
GLANCE		**74.0**	**58.7**	<u>55.3</u>	**62.7**
VisionZip	10%	65.7	54.2	<u>53.8</u>	57.9
FastVID		67.1	53.6	50.9	57.2
FlashVID		69.4	<u>56.3</u>	52.9	<u>59.6</u>
HoliTom		66.9	52.0	51.3	56.7
V-CAST		<u>70.8</u>	54.6	53.6	<u>59.6</u>
GLANCE		**72.9**	**58.7**	**54.1**	**61.9**

A.2 LLaVA-Video-7B

Table 6 reports the corresponding sweep with 64 frames. GLANCE attains the highest mean at 10%, 15%, and 20% retention. At 25%, FlashVID obtains a slightly higher mean (62.1 versus 61.9).

Table 6: Comparison on LLaVA-Video-7B (64 frames). Accuracy and relative performance are percentages; TTFT is in milliseconds. Within each retention group, bold and underline mark the best and second-best distinct displayed values among compression methods, including ties.

Method	r	VMME	MVB	LVB	MLVU	Mean	Rel.	TTFT
Full tokens	100%	63.8	61.0	59.2	66.2	62.6	100.0	760.1
VisionZip	25%	60.8	59.8	58.6	64.8	61.0	97.5	496.9
FastVID		62.2	**60.6**	58.3	66.1	61.8	98.8	492.4
FlashVID		**62.6**	60.5	58.8	66.5	**62.1**	**99.2**	470.9
HoliTom		62.0	60.2	58.0	**66.6**	61.7	98.6	535.2
V-CAST		61.1	59.8	58.6	63.7	60.8	97.2	430.0
GLANCE		62.1	60.3	**59.1**	66.3	61.9	99.0	**415.9**
VisionZip	20%	59.9	59.4	58.5	64.7	60.6	96.9	461.6
FastVID		61.6	60.4	57.4	65.1	61.1	97.7	456.0
FlashVID		61.7	60.0	58.4	65.5	61.4	98.1	451.0
HoliTom		61.4	60.2	58.4	**66.2**	61.6	98.4	507.8
V-CAST		60.1	59.5	57.3	62.7	59.9	95.7	408.1
GLANCE		**62.1**	**60.8**	**59.6**	65.8	**62.1**	**99.2**	**394.9**
VisionZip	15%	58.9	59.2	57.7	63.0	59.7	95.4	425.9
FastVID		61.3	60.0	57.1	64.4	60.7	97.0	429.4
FlashVID		61.8	60.2	58.9	65.5	61.6	98.4	427.8
HoliTom		60.3	59.7	57.4	**66.2**	60.9	97.3	479.3
V-CAST		59.0	58.1	55.1	61.2	58.3	93.3	386.5
GLANCE		**62.7**	**60.7**	**60.1**	65.5	**62.2**	**99.5**	**371.6**
VisionZip	10%	58.1	58.5	56.6	61.9	58.8	94.0	382.7
FastVID		59.5	60.0	55.9	62.7	59.5	95.1	397.8
FlashVID		61.0	59.4	57.4	64.6	60.6	96.9	411.3
HoliTom		58.8	59.4	57.3	64.0	59.9	95.7	449.3
V-CAST		56.9	55.0	52.6	58.5	55.8	89.1	364.5
GLANCE		**61.4**	**60.1**	**57.8**	**64.9**	**61.0**	**97.6**	**350.2**

Table 7: VideoMME duration breakdown for the configurations in Table 6. Scores are percentages.

Method	r	Short	Medium	Long	All
Full tokens	100%	76.3	61.3	53.7	63.8
VisionZip	25%	73.2	58.8	50.4	60.8
FastVID		**75.0**	60.2	51.3	62.2
FlashVID		74.3	**61.4**	**51.9**	**62.6**
HoliTom		74.4	60.4	51.0	62.0
V-CAST		73.4	58.2	51.8	61.1
GLANCE		74.8	59.7	51.8	62.1
VisionZip	20%	71.7	58.1	49.8	59.9
FastVID		74.0	59.8	51.0	61.6
FlashVID		73.3	59.3	**52.6**	61.7
HoliTom		73.9	59.4	50.9	61.4
V-CAST		71.1	57.8	51.3	60.1
GLANCE		**74.1**	**59.9**	52.2	**62.1**
VisionZip	15%	71.2	56.0	49.3	58.9
FastVID		73.0	59.6	51.2	61.3
FlashVID		**73.8**	59.9	51.8	61.8
HoliTom		73.3	58.7	48.8	60.3
V-CAST		69.8	56.3	50.9	59.0
GLANCE		73.6	**61.7**	**52.8**	**62.7**
VisionZip	10%	69.0	56.8	48.7	58.1
FastVID		70.1	57.9	**50.6**	59.5
FlashVID		72.8	59.9	50.2	61.0
HoliTom		71.9	56.6	47.9	58.8
V-CAST		66.2	54.9	49.6	56.9
GLANCE		**73.0**	**60.8**	50.4	**61.4**

A.3 QWEN3.8-27B

Table 8 reports the 32-frame sweep on Qwen3.8-27B, comparing VisionZip, FastVID, VidCom^2, V-CAST, and GLANCE with the full-token reference. GLANCE attains the highest four-benchmark mean among the compression methods at every evaluated retention rate.

Table 8: Comparison on Qwen3.8-27B (32 frames). Accuracy and relative performance are percentages; TTFT is in milliseconds. Within each retention group, bold and underline mark the best and second-best distinct displayed values among compression methods, including ties.

Method	r	VMME	MVB	LVB	MLVU	Mean	Rel.	TTFT
Full tokens	100%	68.7	71.2	63.6	65.1	67.2	100.0	2635.4
VisionZip		<u>66.0</u>	68.9	<u>63.0</u>	**63.8**	<u>65.4</u>	<u>97.4</u>	998.1
FastVID		62.7	67.4	59.5	61.0	62.6	93.3	**896.3**
VidCom^2	25%	**66.1**	68.8	61.4	62.2	64.6	96.2	911.1
V-CAST		63.6	<u>69.1</u>	61.8	<u>63.7</u>	64.5	96.1	903.8
GLANCE		65.6	**70.4**	**63.4**	**63.8**	**65.8**	**97.9**	<u>902.7</u>
VisionZip		<u>65.6</u>	67.9	<u>62.1</u>	**63.7**	<u>64.8</u>	<u>96.5</u>	911.4
FastVID		62.3	66.8	59.5	61.6	62.6	93.1	842.3
VidCom^2	20%	64.7	67.9	59.2	62.1	63.5	94.5	<u>838.2</u>
V-CAST		62.9	<u>68.1</u>	61.8	63.2	64.0	95.3	842.5
GLANCE		**65.9**	**69.5**	**63.0**	<u>63.5</u>	**65.5**	**97.5**	**818.8**
VisionZip		63.6	65.5	<u>61.0</u>	**63.3**	<u>63.3</u>	<u>94.3</u>	818.4
FastVID		60.6	66.6	59.2	60.5	61.7	91.9	774.7
VidCom^2	15%	<u>64.3</u>	66.3	58.9	60.2	62.4	92.9	**750.3**
V-CAST		61.9	<u>67.8</u>	60.8	<u>63.0</u>	<u>63.3</u>	<u>94.3</u>	759.1
GLANCE		**64.7**	**68.7**	**62.3**	**63.3**	**64.7**	**96.4**	<u>751.9</u>
VisionZip		<u>61.8</u>	61.4	59.3	61.2	60.9	90.7	737.0
FastVID		59.9	<u>65.8</u>	57.4	61.0	61.0	90.9	698.3
VidCom^2	10%	61.7	63.4	56.7	58.1	60.0	89.3	**673.1**
V-CAST		60.7	65.6	<u>60.3</u>	**62.3**	<u>62.2</u>	<u>92.6</u>	682.1
GLANCE		**63.7**	**66.6**	**61.1**	<u>61.8</u>	**63.3**	**94.2**	<u>676.9</u>

Table 9: VideoMME duration breakdown for the configurations in Table 8. Scores are percentages.

Method	r	Short	Medium	Long	All
Full tokens	100%	79.4	67.9	58.9	68.7
VisionZip	25%	**76.0**	64.7	**57.4**	66.0
FastVID		71.7	62.2	54.3	62.7
VidCom^2		75.8	**66.1**	56.3	**66.1**
V-CAST		74.1	62.6	54.1	63.6
GLANCE		75.7	65.1	56.0	65.6
VisionZip	20%	76.0	64.4	56.2	65.6
FastVID		69.9	61.6	55.4	62.3
VidCom^2		73.7	63.3	**57.2**	64.7
V-CAST		71.3	62.4	54.9	62.9
GLANCE		**76.1**	**65.3**	56.2	**65.9**
VisionZip	15%	72.1	63.1	**55.7**	63.6
FastVID		68.6	59.9	53.4	60.6
VidCom^2		73.8	63.7	55.3	64.3
V-CAST		70.0	61.8	53.8	61.9
GLANCE		**75.1**	**63.8**	55.1	**64.7**
VisionZip	10%	69.4	61.0	55.0	61.8
FastVID		67.9	58.6	53.2	59.9
VidCom^2		70.0	61.2	53.8	61.7
V-CAST		69.0	60.0	53.0	60.7
GLANCE		**73.1**	**62.7**	**55.2**	**63.7**

A.4 Accuracy–Latency Comparisons Across Backbones

Figure 3 complements the fixed-retention comparison with the measured accuracy–latency curves over all four retention settings. For Qwen3-VL, GLANCE at 25% obtains a mean score of 63.2 at 276.4 ms, compared with FlashVID at 10%, which obtains 59.9 at 284.7 ms. For LLaVA-Video, GLANCE at 20% obtains 62.1 at 394.9 ms, compared with FlashVID at 10%, which obtains 60.6 at 411.3 ms. In both examples, GLANCE processes a larger visual budget at lower measured TTFT. The curves connect measured configurations with method-specific computational costs.

A.5 Frame-Scaling Results

Table 10 gives the complete per-method numerical results underlying Figure 4 across the tested frame counts. The experiment uses Qwen3-VL-8B-Instruct with sampled-frame budgets $T \in \{64, 128, 256\}$, retaining the pixel-budget settings, FlashAttention-2 implementation, greedy decoding, and eight-worker batch-size-one protocol described in Appendix C. Each frame count has its own decoded-frame cache and full-token reference. All methods share the three task keys `videomme_long`, `longvideobench_val_v`, and `mlvu_dev`, with 900, 1,337, and 2,174 effective evaluation questions, respectively; no evaluation limit is applied. The retention rate is fixed at $r = 10\%$, and GLANCE uses the same 1.25 input expansion and final-quarter withdrawal as in the main comparison.

The three-task mean and its relative performance are

$$A_{\text{mean},3} = \frac{A_{\text{VMME-long}} + A_{\text{LVB}} + A_{\text{MLVU}}}{3}, \qquad A_{\text{rel},3}(T) = 100\frac{A_{\text{mean},3}(T)}{A^{\text{full}}_{\text{mean},3}(T)}. \tag{6}$$

These averages use VideoMME's long-duration subset and exclude MVBench, so they differ from the four-benchmark averages in the main retention sweep. TTFT and TPOT use the within-run worker aggregation in Appendix C.3; peak memory is the maximum allocated PyTorch memory across workers, including model weights. Figure 4 compares six compression methods across frame counts and highlights 128-frame GLANCE against the 64-frame full-token reference; the caption reports the increase in peak memory. Table 10 also gives the full-token results at 128 and 256 frames.

Table 10: Complete frame-scaling results on Qwen3-VL at $r = 10\%$, with full-token references at $r = 100\%$. VMME-L uses only the 900 long-duration questions. Mean_3 is the equal-weight mean of the three scores. TTFT is in ms, TPOT in ms/token, and memory is peak PyTorch allocated memory in GiB. Bold and underline mark the best and second-best distinct displayed values among compression methods at each frame count, including ties. Higher accuracy and Rel., and lower TTFT, TPOT, and memory are better.

Frames	Method	VMME long	LVB	MLVU	Mean_3	Rel.	TTFT	TPOT	Memory
64	Full tokens	56.0	62.5	68.9	62.4	100.0	1383.0	37.3	23.6
64	VisionZip	52.7	57.0	59.7	56.5	90.4	624.5	37.3	<u>22.7</u>
64	FastVID	53.4	58.1	61.7	57.8	92.5	518.0	**36.9**	**21.9**
64	FlashVID	**55.2**	58.9	61.6	<u>58.6</u>	<u>93.8</u>	609.1	<u>37.0</u>	**21.9**
64	HoliTom	51.0	56.8	60.0	55.9	89.6	583.6	37.5	<u>22.7</u>
64	V-CAST	52.2	**60.0**	<u>61.8</u>	58.0	92.9	**505.6**	37.1	**21.9**
64	GLANCE	<u>55.0</u>	<u>59.0</u>	**64.4**	**59.5**	**95.2**	<u>508.9</u>	37.8	**21.9**
128	Full tokens	58.9	65.2	73.1	65.7	100.0	3431.3	39.0	30.8
128	VisionZip	55.3	59.7	64.5	59.8	91.1	1273.4	37.7	<u>29.0</u>
128	FastVID	55.6	61.2	65.7	60.8	92.6	1010.1	38.0	**27.4**
128	FlashVID	56.4	61.8	65.9	61.4	93.4	1188.9	**37.2**	**27.4**
128	HoliTom	53.4	58.5	63.7	58.5	89.1	1189.6	38.0	<u>29.0</u>
128	V-CAST	<u>57.6</u>	<u>61.9</u>	<u>66.9</u>	<u>62.1</u>	<u>94.5</u>	<u>1000.5</u>	37.6	**27.4**
128	GLANCE	**57.8**	**62.6**	**68.7**	**63.0**	**95.9**	**981.3**	<u>37.5</u>	**27.4**
256	Full tokens	60.4	65.8	75.9	67.4	100.0	9730.1	43.0	45.3
256	VisionZip	57.0	61.5	68.5	62.3	92.5	2962.0	38.4	42.0
256	FastVID	55.7	61.6	<u>69.8</u>	62.4	92.6	<u>2031.2</u>	37.9	**38.3**
256	FlashVID	<u>58.4</u>	61.2	69.3	<u>63.0</u>	<u>93.5</u>	2403.4	<u>37.8</u>	**38.3**
256	HoliTom	54.7	60.5	66.3	60.5	89.8	2552.1	37.9	<u>41.7</u>
256	V-CAST	57.4	<u>62.4</u>	69.1	<u>63.0</u>	<u>93.5</u>	2049.0	38.7	**38.3**
256	GLANCE	**58.9**	**63.1**	**72.4**	**64.8**	**96.1**	**2010.5**	**37.6**	**38.3**

B Additional Ablation Studies

We examine the scoring and selection components and visual-token withdrawal schedule. These experiments expand the main ablations in Sections 4.3 and 4.5.

B.1 Per-Benchmark Selection Ablations

Random and uniform selection replace IUS and PRS with direct per-group selection of k_f tokens; uniform selection uses evenly spaced positions. Removing informativeness retains global uniqueness alone, while removing global uniqueness retains feature-norm rank alone, with PRS unchanged. Removing PRS selects the top k_f IUS scores directly. All variants keep the full method's input budget and withdrawal schedule.

Qwen3-VL. Table 11 expands the aggregate results in Table 3(a) with individual benchmark scores and VideoMME duration breakdowns. The feature-norm-rank-only variant attains a four-benchmark mean of 61.0 at 234.8 ms; combining informativeness with global uniqueness raises the mean to 61.4. The saved norm-only predictions supply the per-benchmark results and the VideoMME duration breakdown in the table. IUS scores higher on VideoMME and MLVU, while norm rank scores higher on MVBench and LongVideoBench; the mean improvement therefore comes with task-level trade-offs.

Table 11: Selection ablations on Qwen3-VL at 10% retention. All rows use the same input budget and depth schedule. Random and uniform selection replace both IUS and PRS. Bold and underline mark the best and second-best distinct displayed accuracy or TTFT in each column, including ties.

Selection	VideoMME				MVB	LVB	MLVU	Mean	TTFT (ms)
	Short	Med.	Long	All					
Random	68.2	56.1	53.3	59.2	64.2	56.0	58.7	59.5	<u>234.0</u>
Uniform	69.4	54.6	52.3	58.8	63.4	56.8	58.0	59.2	**233.6**
Uniqueness + PRS	69.4	56.6	<u>53.9</u>	60.0	62.0	53.4	58.8	58.5	234.8
Norm rank + PRS	69.4	55.4	53.1	59.3	**65.7**	**58.8**	60.3	<u>61.0</u>	234.8
IUS without PRS	<u>70.2</u>	<u>58.1</u>	53.2	<u>60.5</u>	63.1	57.1	<u>60.4</u>	60.3	234.3
GLANCE	**72.9**	**58.7**	**54.1**	**61.9**	<u>65.2</u>	<u>57.6</u>	**61.1**	**61.4**	234.4

LLaVA-Video. Table 12 gives the individual benchmark scores for all six variants summarized in Table 3(b). At 64 frames and $r = 10\%$, the full method attains a mean of 61.0, compared with 60.3 for norm rank with PRS and 60.6 for IUS without PRS. The norm-only variant obtains 60.3 on VideoMME and 56.6 on LongVideoBench, compared with 61.4 and 57.8 for the full IUS score. The four-benchmark mean improves on both backbones. Appendix D gives paired task-level diagnostics on separate cohorts. All additional full-task runs use eight data-parallel workers. For the LLaVA norm-only variant, we first obtain the worker-averaged TTFT of the existing VideoMME/LongVideoBench run and the supplementary MVBench/MLVU run, then combine these two run summaries using their recorded request counts (Appendix C.3).

Table 12: LLaVA-Video selection ablations at 64 frames and 10% retention. All rows use the same input quotas and depth. Bold marks the best displayed accuracy.

Selection	VMME	MVB	LVB	MLVU	Mean
Random	59.7	59.6	56.5	63.2	59.7
Uniform	59.5	57.9	56.2	63.1	59.2
Uniqueness + PRS	57.6	55.2	52.5	61.0	56.6
Norm rank + PRS	60.3	60.4	56.6	63.9	60.3
IUS without PRS	60.7	**60.6**	56.7	64.5	60.6
GLANCE (IUS + PRS)	**61.4**	60.1	**57.8**	**64.9**	**61.0**

B.2 CANDIDATE-POOL SIZE

Table 13 expands Table 3(d) with per-benchmark scores. The scoring rule, output quotas, PRS discount, and withdrawal depth are fixed; only the number of candidates compared within each group changes. At $m_f = k_f$, every candidate is retained, so PRS cannot change the subset. The $2k_f$ pool reaches the same displayed mean as the larger pools at lower measured TTFT in this Qwen3-VL setting.

Table 13: Candidate pools on Qwen3-VL (32 frames, $r = 10\%$). Quotas, PRS, and depth are fixed; All uses every token in each group.

Candidates	VMME	MVB	LVB	MLVU	Mean	TTFT (ms)
k_f	60.5	63.1	57.1	60.4	60.3	234.6
$2k_f$ (default)	61.9	65.2	57.6	61.1	61.4	234.4
$4k_f$	61.2	65.7	58.0	60.8	61.4	240.8
All	61.2	65.7	58.0	60.7	61.4	244.9

B.3 DEPTH-SCHEDULE ABLATIONS

Table 14 distinguishes four schedules with the same IUS score and PRS selector. The full schedule selects $1.25rN$ tokens and withdraws visual positions for the final quarter of decoder layers. The selection-only variant disables both input expansion and visual token removal. The expansion-only variant keeps the larger subset through all layers; the withdrawal-only variant uses the smaller input budget at the same withdrawal depth. Integer budgets are $B = \lceil rN \rceil$ for selection-only and withdrawal-only, and $B = \lceil 1.25rN \rceil$ for expansion-only and the complete configuration.

At 10% retention, the full schedule and expansion-only variant use the same 12.5% input subset and both attain a displayed mean of 61.4. Withdrawal reduces the layer-averaged fraction to 9.4% and TTFT from 242.8 to 234.4 ms, saving 8.4 ms (3.5%). With the smaller 10% input subset, selection alone obtains 61.1 at 233.1 ms, and withdrawal alone obtains 61.0 at 230.7 ms. The full schedule thus adds 1.3 ms over selection alone for a 0.3-point gain in displayed mean. At 20% retention, the full schedule obtains 63.0 mean accuracy at 261.2 ms, compared with 62.9 at 273.2 ms for expansion only. Withdrawal saves 12.0 ms (4.4%) in this setting. These comparisons quantify the accuracy–latency trade-off of input expansion and withdrawal at the default depth.

Table 14: Input-budget and processing-depth ablations at 10% and 20% retention on Qwen3-VL. B/N is the input fraction and $\bar{r}$ the layer-averaged fraction, ignoring integer rounding. Selection is otherwise unchanged. VideoMME reports All. Bold and underline mark the best and second-best distinct displayed accuracy or TTFT within each retention setting, including ties; budget columns are configuration values. TTFT differences are single-run measurements.

Schedule	Visual budget			Benchmark accuracy				Mean	TTFT (ms)
	B/N	K/L	$\bar{r}$	VMME	MVB	LVB	MLVU		
				$r = 10\%$					
Selection only	10%	1	10%	<u>61.4</u>	64.5	<u>57.5</u>	<u>61.0</u>	<u>61.1</u>	<u>233.1</u>
Expansion only	12.5%	1	12.5%	**61.9**	<u>65.1</u>	**57.6**	**61.1**	**61.4**	242.8
Withdrawal only	10%	3/4	7.5%	<u>61.4</u>	64.5	57.4	60.8	61.0	**230.7**
GLANCE	12.5%	3/4	9.4%	**61.9**	**65.2**	**57.6**	**61.1**	**61.4**	234.4
				$r = 20\%$					
Selection only	20%	1	20%	62.7	<u>66.9</u>	<u>58.2</u>	61.6	62.3	<u>259.2</u>
Expansion only	25%	1	25%	**63.1**	**68.0**	**58.6**	<u>62.0</u>	<u>62.9</u>	273.2
Withdrawal only	20%	3/4	15%	<u>62.9</u>	66.8	58.0	61.6	62.3	**250.2**
GLANCE	25%	3/4	18.8%	**63.1**	**68.0**	**58.6**	**62.1**	**63.0**	261.2

B.4 Withdrawal Depth at a Fixed Token–Layer Budget

We vary the number of layers processing visual tokens on Qwen3-VL with 32 frames ($L = 36$) and LLaVA-Video with 64 frames ($L = 28$). For each K, we set $B/N = 0.09375L/K$, keeping $(B/N)(K/L) = 0.09375$ up to integer token rounding. This aligns the layer-averaged visual-token budget, not FLOPs, since the input sequence lengths differ.

Table 15 reports the results.

Table 15: Withdrawal-layer sensitivity at a fixed layer-averaged visual-token fraction of 9.375%. K layers process visual tokens; B/N is the input fraction. † marks the default configuration reported in Table 1. TTFT is in ms.

K	K/L	B/N (%)	VMME	MVB	LVB	MLVU	Mean	TTFT
Qwen3-VL, 32 frames								
9	9/36	37.5	44.6	42.9	45.7	45.7	44.7	245.9
18	18/36	18.8	59.7	62.1	57.4	60.8	60.0	239.8
25	25/36	13.5	62.3	65.0	57.6	60.8	61.4	236.7
26	26/36	13.0	62.0	65.1	57.3	61.0	61.4	235.1
27†	27/36	12.5	61.9	65.2	57.6	61.1	61.4	234.4
28	28/36	12.1	61.6	65.3	56.9	61.2	61.3	234.3
29	29/36	11.6	61.1	65.0	56.9	60.7	60.9	234.0
LLaVA-Video, 64 frames								
7	7/28	37.5	43.9	41.5	43.9	46.2	43.9	388.5
14	14/28	18.8	53.6	51.6	49.8	55.8	52.7	362.2
19	19/28	13.8	61.7	60.3	58.1	65.1	61.3	353.2
20	20/28	13.1	61.4	59.7	58.1	65.0	61.0	352.5
21†	21/28	12.5	61.4	60.1	57.8	64.9	61.0	350.2
22	22/28	11.9	61.4	60.1	57.9	64.7	61.0	349.7
23	23/28	11.4	61.3	59.9	57.1	64.7	60.8	347.9

Early withdrawal reduces accuracy even with the larger input budget: at $K/L = 1/4$, the four-benchmark mean falls to 44.7 on Qwen3-VL and 43.9 on LLaVA-Video. Over the tested late-layer ranges, $K = 25$–29 and $K = 19$–23, the displayed means span 60.9–61.4 and 60.8–61.3, respectively. The Qwen3-VL default ($K = 27$) ties for the highest displayed mean. On LLaVA-Video, $K = 19$ scores 61.3 versus 61.0 at the default $K = 21$, with slightly higher measured TTFT (353.2 versus 350.2 ms).

For LLaVA-Video at $K = 19$, an independent full rerun reproduced all 10,211 predictions and measured 353.2 ms TTFT. The initial run measured 4685.4 ms TTFT with long stalls; this anomaly did not recur and remains unexplained. We report the rerun timing and retain both records.

C Evaluation and Implementation Details

This section specifies the benchmarks, shared inference protocol, reporting conventions, and implementation of GLANCE. It also gives the token-budget accounting and the matched selector profiling protocol used to interpret the results in Appendices A and B.

C.1 Benchmarks and Compared Methods

We evaluate four video-understanding benchmarks using benchmark-specific scoring and answer extraction in our lmms-eval implementation:

- **VideoMME (VMME)** (Fu et al., 2025a): 2,700 questions without subtitles, using task key `videomme`. Short, Medium, and Long each contain 900 questions; their scores are computed from saved predictions and duration labels. The All score enters the four-benchmark mean.

- **MVBench (MVB)** (Li et al., 2024b): 4,000 questions, using task key `mvbench`.
- **LongVideoBench (LVB)** (Wu et al., 2024): 1,337 questions from the video-only validation split, using task key `longvideobench_val_v`.
- **MLVU** (Zhou et al., 2025): 2,174 questions from the Multi-task Long Video Understanding development multiple-choice tasks, using task key `mlvu_dev`.

We evaluate Qwen3-VL-8B-Instruct (Bai et al., 2025) and Qwen3.8-27B (Qwen Team, 2026) with 32 frames, and LLaVA-Video-7B (Zhang et al., 2024b) with 64 frames. The checkpoint identifiers are `Qwen/Qwen3-VL-8B-Instruct`, `Qwen/Qwen3.8-27B`, and `lmms-lab/LLaVA-Video-7B-Qwen2`, respectively; each uses its associated pretrained processor. For Qwen3-VL and LLaVA-Video, the compression baselines are VisionZip (Yang et al., 2025), FastVID (Shen et al., 2026), FlashVID (Fan et al., 2026), HoliTom (Shao et al., 2026), and V-CAST (Lin et al., 2026). The Qwen3.8-27B comparison includes VisionZip, FastVID, VidCom^2 (Liu et al., 2025), V-CAST, and GLANCE. All baseline scores come from local evaluation under the shared pipeline at retention rates $r \in \{10, 15, 20, 25\}\%$, with a full-token reference for each backbone. The retention rate specifies a configured compression level; actual input and layer-averaged token counts follow the method's depth schedule (Appendix C.5).

C.2 Visual Input and Inference Protocol

Qwen3-VL uses 32 decoded frames with the processor's video sampling and tensor resizing, configured with `min_pixels=200704` and `max_pixels=12845056`. The processors determine the final spatial resolution from these settings. LLaVA-Video uses 64 frames with forced uniform sampling, its pretrained image processor, average spatial pooling with stride two, the `qwen_1_5` conversation template, frame-level newline tokens, and no additional time instruction. Qwen3-VL and LLaVA-Video use FlashAttention-2 (Dao, 2024). Qwen3.8-27B uses 32 frames with the same pixel-budget settings as Qwen3-VL and its own pretrained processor. Frames are decoded on the CPU with Decord and cached on node-local storage before evaluation. Within each backbone, all methods reuse the same RGB frame cache and image-processing configuration. Frame directories are read in filename order; the decoder retains the existing PyAV fallback for codecs unsupported by Decord. The cache stores video frames, not answers or question-conditioned representations.

The evaluation uses greedy generation, the same task prompts and stopping rules within each backbone, and batch size one on each of eight data-parallel workers. For Qwen3-VL and LLaVA-Video, the generation limits are 16 tokens for Video-MME, MVBench, and MLVU, and 32 for LongVideoBench. For Qwen3.8-27B, the limit is 4,096 tokens for every benchmark, thinking is disabled, and the shared system prompt requests exactly one uppercase option letter without reasoning or repeated question text.

C.3 Metrics and Timing

For benchmark scores A_d on the percentage scale, we report

$$A_{\text{mean}} = \frac{1}{4}\sum_{d=1}^{4} A_d, \qquad A_{\text{rel}} = 100\frac{A_{\text{mean}}}{A_{\text{mean}}^{\text{full}}}. \tag{7}$$

Each benchmark receives equal weight; relative performance is the ratio of these means. Unless otherwise stated, means and relative accuracies are computed from unrounded benchmark scores and rounded only for display; recomputing them from the displayed scores can therefore yield a different last digit. For the ablation variants in Table 3(a,b), relative accuracies are computed from the displayed means and full-token references (64.4 for Qwen3-VL and 62.6 for LLaVA-Video), then rounded to one decimal place; the GLANCE rows retain the relative accuracies reported in Table 1. For Table 3(c), the three-task mean is computed from the displayed task scores. Accuracy changes denote absolute differences on the 0–100 scale; relative gains divide this difference by the reference score. Where table entries are highlighted, **bold** marks the best displayed values and underline marks the second-best distinct values; ties receive the same formatting. Compression comparisons rank methods within each backbone and retention or frame-count group, excluding the full-token reference. Ablations rank the compared compression variants, excluding full-token references where present.

Algorithm 1 IUS–PRS token selection

Require: Features $V = \{\mathbf{v}_{f,p}\}$, input budget B; normalization $\mathcal{R}$ in Equation (3)

Allocate quotas k_f uniformly with $\sum_f k_f = B$; initialize $S_f \leftarrow \emptyset$ for all f
$I_{f,p} \leftarrow \mathrm{rank}_p(\|\mathbf{v}_{f,p}\|_2)/(P-1)$
$\hat{V} \leftarrow \mathrm{NormalizeRows}(V)$; $\mathbf{c}_g \leftarrow N^{-1} \sum_{f,p} \hat{\mathbf{v}}_{f,p}$
$g_{f,p} \leftarrow 1 - \hat{\mathbf{v}}_{f,p}^\top \mathbf{c}_g$; $G \leftarrow \mathcal{R}(g)$
$U \leftarrow \mathcal{R}(I \odot G)$
for all groups f with $k_f > 0$ **in parallel** (chunked) **do**
    $C_f \leftarrow \mathrm{TopK}(U[f,:], \min(2k_f, P))$ in returned descending-utility order
    $X \leftarrow \hat{V}[C_f]$; $u \leftarrow U[C_f]$; $A \leftarrow XX^\top$
    $A_{ji} \leftarrow 0$ for $i \geq j$ ▷ mask self and later candidates
    $\rho_j \leftarrow \max(0, \max_i A_{ji})$ for all j in parallel
    $s_j \leftarrow u_j(1 - \rho_j)$ for all j in parallel
    $S_f \leftarrow C_f[\mathrm{StableTopK}(s, k_f)]$
end for
$S \leftarrow \mathrm{SortOriginalOrder}(\bigcup_f S_f)$
return $V[S]$ and original position indices S

Higher accuracy and Rel. are better, while lower TTFT, TPOT, and memory are better; configuration columns are not ranked.

TTFT starts at `model.generate` and ends when the first generated token reaches the latency streamer. It includes visual encoding, compression, and the first decoder forward pass, while frame-cache I/O and CPU preprocessing occur before the timer starts. CUDA is synchronized before generation. Each full-benchmark run uses eight data-parallel workers with batch size one. Within run q, worker w reports its mean request TTFT $\bar{t}_{q,w}$; the evaluator averages these worker reports equally:

$$\bar{t}_q = \frac{1}{W} \sum_{w=1}^{W} \bar{t}_{q,w}, \qquad W = 8. \tag{8}$$

Single-run results, including frame scaling, use this saved run mean. When a configuration's benchmarks are split across runs, the supplementary aggregation script combines the run means using each run's recorded request count n_q:

$$\bar{t}_{\mathrm{combined}} = \frac{\sum_q n_q \bar{t}_q}{\sum_q n_q}. \tag{9}$$

Request-count weighting therefore applies between runs, after equal-weight worker aggregation within each run; it is not a pooled request-weighted average across all workers. TPOT uses the same aggregation hierarchy for the available worker reports. The separate selector-profiling experiment instead averages 150 request-level TTFT measurements per configuration (30 videos, five rounds; Appendix C.6). Peak memory is the maximum PyTorch-allocated memory across workers, including model weights but excluding external allocations.

C.4 Selection Details

IUS operates on the visual embeddings prepared for the decoder. On Qwen3-VL these are the vision tower's spatially merged outputs; on LLaVA-Video they follow multimodal projection and stride-two spatial pooling, before newline insertion. On Qwen3.8-27B, selection uses the spatially merged outputs returned by the vision encoder. We write the encoded video grid as (F, H, W): F is its temporal group count, and each group contains $P = (H/s)(W/s)$ tokens, where s is the spatial merge size. The sampled decoded-frame count remains T. For example, Qwen3-VL's temporal patch size of two maps a $T = 32$ frame input to $F = 16$ groups; zero-based frames 24 and 25 belong to group 12. LLaVA-Video's spatial pooling preserves the temporal dimension, so its $T = 64$ frame input has $F = 64$ groups. The IUS informativeness proxy uses zero-based ascending ranks of the original feature norms within each group. Equal norms receive successive integer ranks in the order returned by `argsort`. The selector computes norms in the input feature dtype and casts feature

directions and response ranks to float32 for scoring. Unit-normalized features are used for global scoring and candidate similarity, while the selected output vectors retain their original magnitudes. Rank scaling is invariant to a common positive rescaling within a group.

The global mean is $\mathbf{c}_g = N^{-1} \sum_{f,p} \hat{\mathbf{v}}_{f,p}$. Substituting this mean into Equation (2) gives

$$g_{f,p} = \frac{1}{N} \sum_{f',p'} \left(1 - \hat{\mathbf{v}}_{f,p}^{\top} \hat{\mathbf{v}}_{f',p'}\right). \tag{10}$$

Each summand is a cosine distance between unit-normalized features. This global score is invariant to token permutations; it measures directional distinctiveness, whereas SURGE uses prediction-based temporal surprise (Tang et al., 2026). Min–max normalization in Equation (3) uses the score range over all video tokens. The outer normalization preserves the ordering of the product $I \odot G$.

Each group receives $\lfloor B/F \rfloor$ tokens, with one additional token assigned to each of the first $B \bmod F$ groups in temporal order. Groups with zero budget are skipped. PRS compares each candidate with all predecessors in the initial utility order. Negative similarities contribute zero penalty, and the first candidate's penalty is zero. The initial shortlist is produced by `topk`, which does not guarantee original-position order for equal IUS utilities. Discounted-score ties retain the order returned by that shortlist through a stable sort. The selected indices are then returned in their original temporal and spatial order. Candidate similarities are computed with batched matrix multiplication and a strict lower-triangular mask. The batches are chunked over groups to bound temporary memory. A candidate can be penalized for resembling a predecessor that is later discarded, because the reference set remains fixed.

At the visual token removal boundary, attention masks and position indices are gathered to match the surviving text and structural positions. Attention layers before the removal boundary retain visual KV entries during autoregressive decoding; attention layers after it cache only surviving positions. This description applies to attention KV caches; recurrent layers in a hybrid decoder maintain their own recurrent states. For Qwen3-VL, multi-axis position indices and DeepStack features follow the selected visual indices. On Qwen3.8-27B, the adapter gathers the original multi-axis position indices after selection and gathers hidden states, rotary-position features, and attention masks again at withdrawal. Its hybrid decoder retains per-layer attention KV caches and recurrent states across generation steps. Selection indices depend only on the video features and may be reused across questions, while question-dependent decoder states require fresh computation.

C.5 Input and Layer-Averaged Budgets

Let r be the retention rate and L the number of decoder layers. The evaluated configuration uses

$$\begin{aligned} &B = \lceil 1.25rN \rceil, \\ &K = \lfloor 0.75L \rfloor, \qquad n_v^{(\ell)} = \begin{cases} B, & 1 \le \ell \le K, \\ 0, & K < \ell \le L, \end{cases} \end{aligned} \tag{11}$$

where $n_v^{(\ell)}$ counts the explicit visual feature positions processed by layer ℓ. All selected visual tokens are processed in the first K layers and removed before layer $K+1$. Text, timestamps, and structural separators remain according to the backbone's packing convention and are excluded from the visual-token count. Their representations continue through the remaining layers after earlier cross-modal interaction.

We report both the input fraction B/N and the layer-averaged visual retention fraction

$$\bar{r} = \frac{1}{LN} \sum_{\ell=1}^{L} n_v^{(\ell)} = \frac{KB}{LN}. \tag{12}$$

Ignoring integer rounding in the reported retention range, these fractions are $1.25r$ and $0.9375r$, respectively. At $r = 10\%$, about 12.5% of visual tokens enter the LLM and remain for the first 75% of its layers, yielding $\bar{r} \approx 9.4\%$. Token–layer accounting does not imply equal FLOPs or latency across methods; efficiency comparisons use measured TTFT. This layer-averaged quantity is also used to account for hybrid compression by FlashVID (Fan et al., 2026).

Qwen3-VL-8B has $L = 36$ and $K = 27$; LLaVA-Video-7B has $L = 28$ and $K = 21$; Qwen3.8-27B has $L = 64$ and $K = 48$. The retention rates of 10%, 15%, 20%, and 25% therefore correspond to layer-averaged fractions of approximately 9.4%, 14.1%, 18.8%, and 23.4%. The factor 1.25 is fixed for the default GLANCE configuration. The schedule ablations in Appendices B.3 and B.4 vary input expansion or the input-budget–depth combination. The reported GLANCE configuration passes `glance_stage1_score=global_only` to the FlashVLM implementation: this mode still multiplies the norm-rank score by global uniqueness, as in Equation (4), but excludes the optional local-uniqueness branch and therefore uses no global/local mixing coefficient.

For FlashVID, if a fraction η of the input visual tokens remains after its pruning boundary, the corresponding approximation is

$$\bar{r}_{\mathrm{FlashVID}} = \frac{B}{N}\frac{K + \eta(L - K)}{L}. \tag{13}$$

The recorded Qwen3-VL configuration uses $B/N \simeq 1.25r$, $K = 28$, $L = 36$, and $\eta = 0.1$; the LLaVA-Video configuration uses $K = 20$, $L = 28$, and $\eta = 0.3$. These yield $\bar{r}_{\mathrm{FlashVID}} \simeq r$. Consequently, the two evaluated schedules have similar, but not identical, layer-averaged visual-token budgets: GLANCE uses approximately 6.3% fewer visual token–layer positions at the same retention rate. Actual rounding and method-specific merging affect the counts, and the quadratic attention term further prevents an inference of equal FLOPs from these averages. All latency comparisons use measured TTFT.

C.6 Matched Selector Profiling Protocol

The matched profiling comparison uses direct Top-k, sequential greedy selection, and PRS with fixed scores, output quotas, and processing depth. Greedy and PRS share the top-$2k_f$ candidate pools and multiplicative discount. After each selection, greedy selection updates the remaining candidates' penalties using their similarities to the newly selected token. It precomputes similarities on the GPU and updates groups in batches without per-step host synchronization. The greedy rule is a matched-score selector that isolates the effect of updating the redundancy reference set.

We profile all three rules with 32- and 128-frame inputs on one NVIDIA H100 80GB GPU with bfloat16 and FlashAttention-2 (Dao, 2024). A fixed cohort contains ten videos from each VideoMME duration group, using the first question of each video. After ten warmup requests per configuration, we measure normal TTFT over five rounds of all 30 videos with shuffled request order, yielding 150 measurements per configuration. TTFT includes visual encoding, compression, and the first decoder forward pass from prepared inputs, excluding CPU preprocessing and frame-cache I/O. A separate instrumented pass over the same videos uses CUDA events to measure candidate selection and the complete compression module; these intervals can include host launch gaps. Table 3(c) and Table 16 report mean module times over 30 videos and mean normal TTFT over all five rounds at 32 and 128 frames, respectively. At 128 frames, candidate selection takes 0.17, 8.87, and 1.07 ms for Top-k, Greedy, and PRS, respectively; complete compression takes 2.65, 11.42, and 3.50 ms. The accuracy comparison in Table 3(c) uses the reported full-benchmark scores at 32 frames; its three-task mean is computed from the displayed task scores. This profiling cohort is separate from the full-benchmark accuracy evaluation and its timing in Table 3(a,b).

Table 16: Matched selector profiling at 128 frames on Qwen3-VL, $r = 10\%$, with the same IUS scores, quotas, and inference schedule. Times are in ms. Select.: candidate selection; Compr.: complete compression. The 32-frame results are in Table 3(c).

IUS +	Select.	Compr.	TTFT
Direct Top-k	**0.17**	**2.65**	**1100.70**
Greedy	8.87	11.42	1107.54
PRS (GLANCE)	1.07	3.50	1100.84

At 128 frames, PRS adds 0.85 ms of compression over Top-k, while paired TTFT differs by only $+0.15$ ms (PRS minus Top-k, from unrounded timings). The paired interval spans zero (Appendix D.2); variation in the other TTFT stages can mask the smaller compression cost of Top-k. Relative to Greedy, PRS saves 6.70 ms of TTFT.

D UNCERTAINTY AND REPEATABILITY

We report paired accuracy diagnostics on separate video cohorts and timing intervals from repeated requests. The accuracy cohorts and the selector profiling cohort are analyzed separately, with videos as the resampling unit.

D.1 PAIRED ACCURACY DIAGNOSTICS

Table 17: Paired accuracy diagnostics on the independently specified 150-video subset of each task. The comparison method is matched-score greedy selection for Qwen3-VL and feature-norm-rank-only scoring with PRS for LLaVA-Video. Scores are percentages; differences and percentile intervals are in accuracy points. These are not confidence intervals for the main-table aggregates.

Backbone	Task	Questions	GLANCE	Baseline	Difference	95% CI
Qwen3-VL	VMME	450	61.56	61.11	+0.44	[-1.11, 2.00]
Qwen3-VL	LVB	263	57.79	56.27	+1.52	[0.00, 3.44]
LLaVA-Video	VMME	450	60.44	57.33	+3.11	[0.89, 5.33]
LLaVA-Video	LVB	263	58.56	55.51	+3.04	[-0.38, 6.44]

We use a prespecified, duration-stratified cohort of 150 videos per task, shared across the two backbones, to diagnose the accuracy differences between the selection variants. The cohort includes all questions from these videos: 450 for VideoMME and 263 for LongVideoBench. The full GLANCE reference was rerun on this cohort and paired with the corresponding questions from the complete matched-score greedy run on Qwen3-VL or feature-norm-rank-only run on LLaVA-Video. Matching uses question IDs, video IDs, duration groups, answer labels, and identical input prompts within each comparison. We recompute correctness from the saved parsed predictions and verify that question-level aggregation reproduces each run's saved score to its reporting precision.

We compute 95% percentile intervals from 10,000 paired bootstrap replicates. Within each task, videos are resampled with replacement within duration groups, keeping each group's video count fixed. All questions from a sampled video enter together, and both methods use the same resampled videos. Each replicate recomputes question-weighted task accuracy, rather than averaging per-video accuracies. The tasks are analyzed separately; no pooled task score or four-benchmark mean is inferred from this cohort. The cohort manifest was fixed before the paired analysis.

Table 17 reports GLANCE minus the comparison method. For Qwen3-VL, the differences are +0.44 points on VideoMME and +1.52 on LongVideoBench; both intervals include zero, with the latter touching zero at its lower endpoint. These diagnostics do not establish superiority or formal equivalence between PRS and greedy selection. For LLaVA-Video, adding global uniqueness improves full-benchmark VideoMME and LongVideoBench accuracy by 1.1 and 1.2 points, respectively. On the separate 150-video cohorts, the paired gains are +3.11 points (95% CI [0.89, 5.33]) and +3.04 points ([-0.38, 6.44]); only the VideoMME interval excludes zero. The intervals are exploratory and are not adjusted for multiple comparisons. Their scope is the separate diagnostic cohort; the full-benchmark aggregates in Table 1 use different evaluation sets.

D.2 TIMING VARIABILITY AND PAIRED DIFFERENCES

Table 18 supplements the mean timings with request medians and the recorded median within-video repeat CV. The five rounds are repeated measurements of the same 30 videos, rather than 150 independent videos. To compare selectors, we pair normal TTFT by video and round, average the five paired differences within each video, then resample videos within each of the three duration groups. We use 10,000 paired bootstrap replicates and report percentile intervals in Table 19; positive differences mean that PRS is faster.

The greedy-minus-PRS intervals are [8.52, 13.40] ms at 32 frames and [3.94, 9.58] ms at 128 frames, supporting the observed TTFT reduction on this cohort. Both Top-k-minus-PRS intervals cross zero, consistent with their close normal-TTFT means. These intervals quantify across-video uncertainty conditional on the measured rounds and hardware, not reproducibility across machines or days. They apply to normal TTFT, not to the separately instrumented module timings.

Table 18: Normal TTFT on the fixed 30-video cohort (150 requests per configuration). Mean and median are in ms. CV is the median within-video coefficient of variation across five rounds, expressed as a percentage.

Frames	Selection	Mean	Median	CV (%)
32	Top-k	289.85	298.30	2.41
32	Greedy	299.77	307.18	3.35
32	PRS	288.75	297.67	2.62
128	Top-k	1100.70	1146.25	1.18
128	Greedy	1107.54	1153.31	1.24
128	PRS	1100.84	1146.48	1.31

Table 19: Paired normal-TTFT differences (ms); positive values favor PRS. Requests are paired by video and round, and intervals resample videos within duration groups after averaging the five paired round differences per video.

Frames	Comparison	Difference	95% CI
32	Greedy − PRS	+11.02	[8.52, 13.40]
32	Top-k − PRS	+1.10	[-0.87, 3.09]
128	Greedy − PRS	+6.70	[3.94, 9.58]
128	Top-k − PRS	-0.15	[-3.81, 3.91]

E QUALITATIVE VISUALIZATIONS

We first visualize how the scoring signals and PRS change the selected positions, then compare token selection and answers across methods on Qwen3-VL and LLaVA-Video. The examples complement the aggregate results in Appendix A.

E.1 SCORING SIGNALS AND REDUNDANCY SUPPRESSION

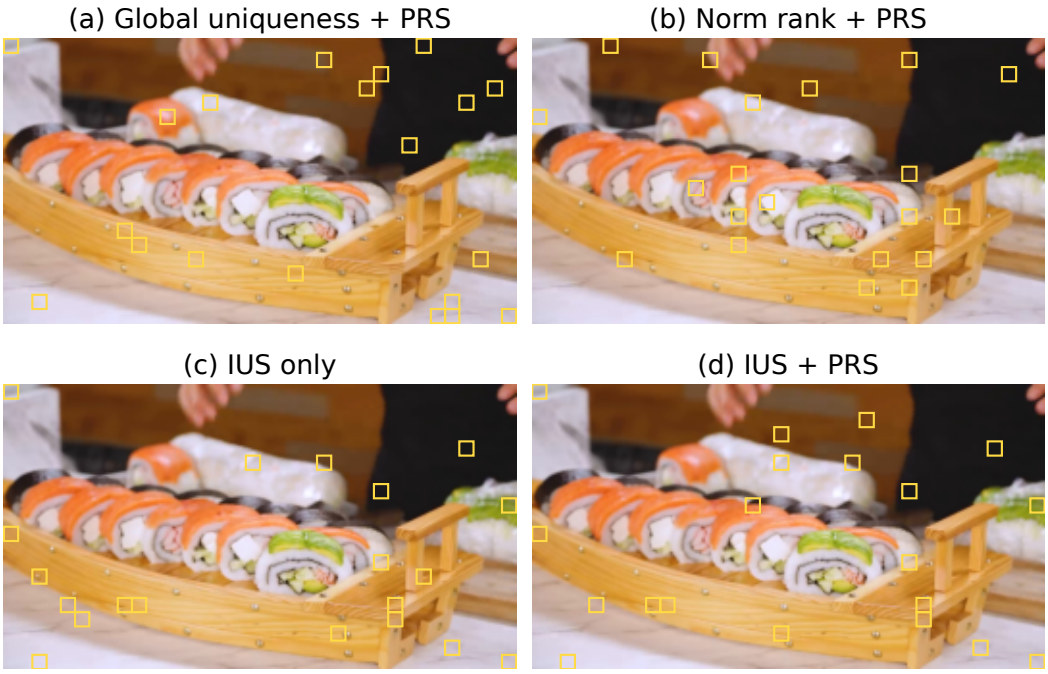


Figure 5: Same-frame selection variants on LongVideoBench video `1R5uPaL0V-0`. Each panel retains 20 of 720 tokens per group for visual clarity, separately from the accuracy-ablation budget. Panel (c) uses direct Top-k; the others use PRS with 40 candidates. Yellow outlines mark retained positions on frame 24, which shares temporal group 12 with frame 25 (zero-based). This diagnostic uses 32 uniformly sampled frames resized to 1152×640, with a 20×36 token grid.

Figure 5 compares two single-signal variants with IUS, with and without PRS, on the same features. This visualization isolates changes in retained positions from the scoring signals and PRS. The video and displayed frame were fixed to the existing qualitative case below; the preprocessing is reconstructed for this comparison and differs from the original compression-trace protocol.

E.2 Score Components and Cross-Method Selection

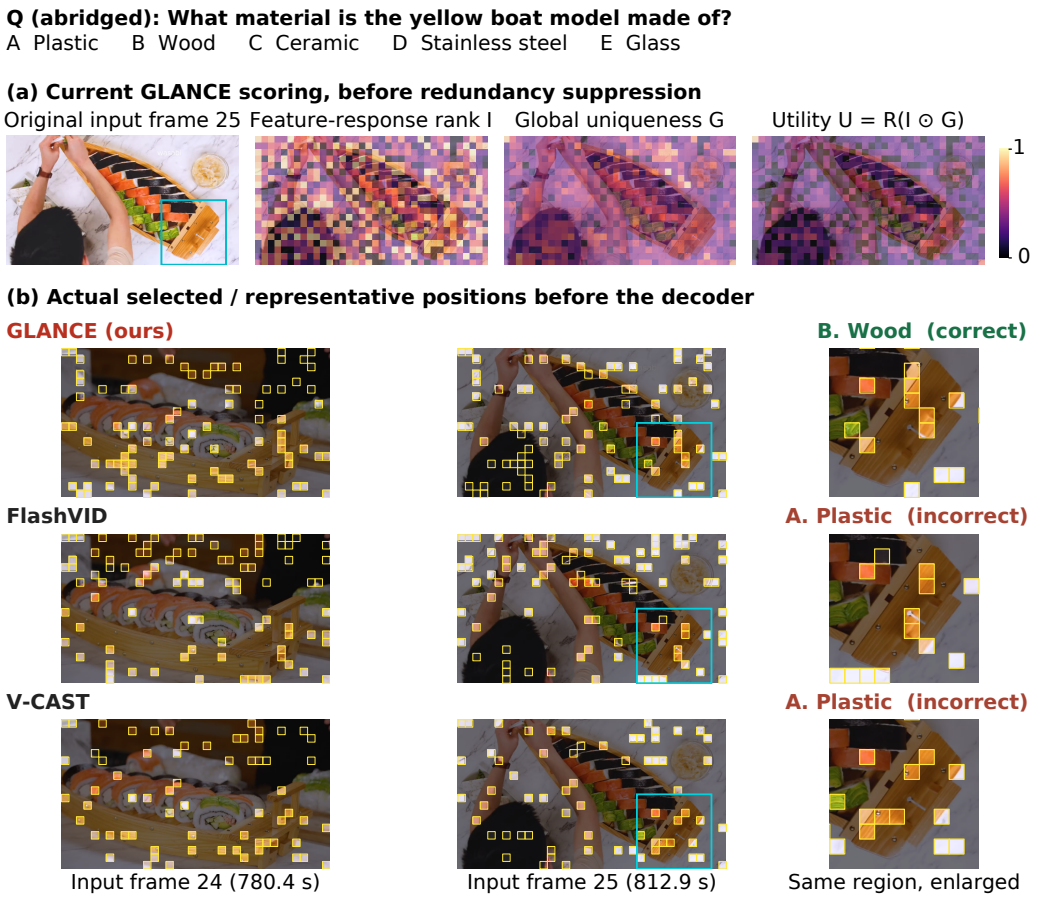


Figure 6: A LongVideoBench case on Qwen3-VL-8B at 32 frames and $r = 10\%$. **(a)** GLANCE scores: feature-norm rank I, global uniqueness G, and normalized product U (Eq. 4). **(b)** Actual pre-decoder token positions, aligned by method; cyan boxes define identical enlarged regions. Input frames 24/25 share temporal token group 12 (zero-based). Yellow outlines show retained positions for GLANCE and V-CAST, and representatives for FlashVID; unmarked positions may contribute to FlashVID's merged tokens. Answers are from reruns with identical inputs; only the displayed question is abbreviated.

We select LongVideoBench question `1R5uPaL0V-0_1`, whose annotated answer is *wood*. Reruns with the evaluated implementations give *wood* for GLANCE and *plastic* for both alternatives. All methods share identical processed videos, prompts, and greedy decoding settings. The paired input frames share a 20×36 spatial grid, with 90, 89, and 66 selected or representative positions for GLANCE, FlashVID, and V-CAST. Their whole-video input counts differ because the first two methods use the configured 1.25 expansion, while V-CAST compresses directly to 10%; FlashVID also applies segmentation and rounding.

This case was chosen from 27 saved questions where GLANCE was correct and both alternatives were incorrect, followed by visual inspection and reruns of four candidates. One candidate was excluded because FlashVID answered correctly on rerun. The figure shows recorded compression-call traces from this outcome-selected case. It illustrates where the methods retain tokens; it does not estimate how often one method preserves better evidence across the benchmark.

E.3 QWEN3-VL: PERSON IDENTIFICATION



Figure 7: Qwen3-VL-8B on LongVideoBench question `kk-iRzLv81o_1`, with 32 frames. GLANCE predicts the annotated answer C; FlashVID and V-CAST predict B, and the full-token baseline predicts A. Reruns use identical prompts, sampled frames, and greedy decoding. Selection outlines follow Figure 6; the full-token row retains all positions. Frame indices are zero-based, with identical cyan crops. Schedules at $r = 10\%$ follow Appendix C.5.

E.4 LLaVA-Video: Object Recognition

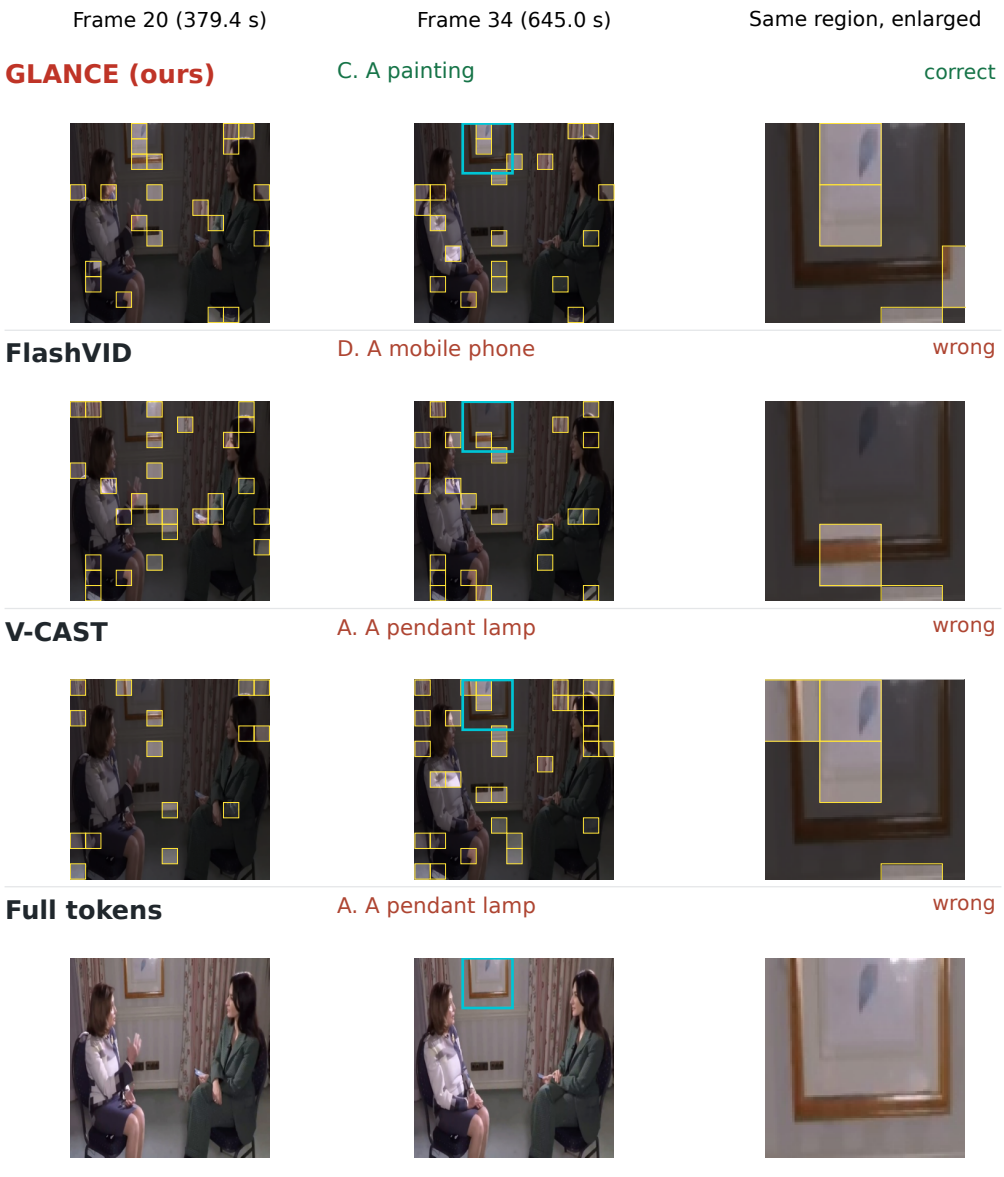


Figure 8: Wall-object recognition on LongVideoBench question `WaiGdRYD36k_1`, using LLaVA-Video-7B and 64 frames. GLANCE correctly predicts C (a painting); FlashVID predicts D (a mobile phone), while V-CAST and the full-token baseline predict A (a pendant lamp). Images show the actual preprocessed inputs, and outlines come from recorded compression-call indices on the 13×13 spatial grid. The rerun protocol, retention, and display conventions follow Figure 7.

E.5 LLaVA-Video: Clothing Recognition

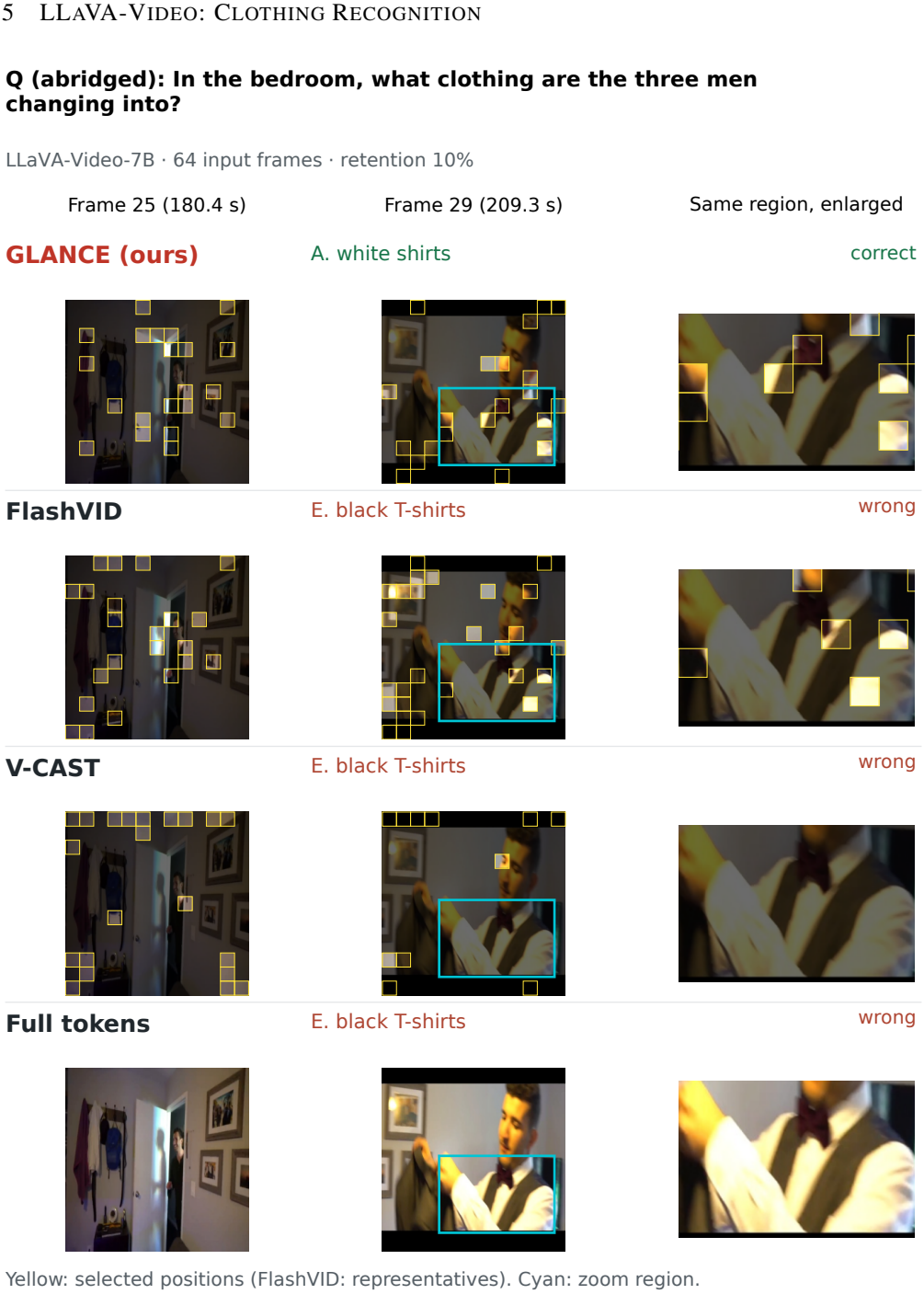


Figure 9: Clothing recognition on LongVideoBench question `qbA42wQoWAs_0`, using LLaVA-Video-7B and 64 frames. GLANCE correctly predicts A (white shirts); FlashVID, V-CAST, and the full-token baseline all predict E (black T-shirts). The inputs, traces, and display conventions follow Figure 8. Figures 7–9 are outcome-selected examples where only GLANCE answers correctly. Displayed questions and long option descriptions are abridged; predictions use the complete evaluation prompts. These examples illustrate individual selection and answer differences; aggregate accuracy is reported in Tables 4 and 6.