PreViC: Predictive Visual Compression for Video Vision-Language Models

Anonymous authors
Paper under double-blind review

Abstract

Video vision-language models (VLMs) turn even a short video into thousands of visual tokens, and those tokens dominate the prefill that precedes every answer. Cutting them before prefill, without retraining the model, is where compression pays off in deployment. Training-free methods for this stage mostly rank a token by how it looks on its own, using saliency, diversity or similarity to neighboring frames. Such a score says nothing about how much a token adds to what has already been kept, so it keeps detail that nearby frames already explain and discards small changes that they do not. We propose **PreViC** (**Pre**dictive **Vi**sual **C**ompression), which keeps a visual token exactly when a compact visual memory fails to predict it. PreViC is training-free and prompt-agnostic, and runs before the language model. A few temporally novel anchor frames form that memory, and every other token is ranked by how poorly its nearest anchors predict it. Over three checkpoints and five retain ratios, PreViC has the highest four-benchmark average accuracy of any method at 12 of the 15 settings. On LLaVA-OneVision 7B it holds the uncompressed model's accuracy on 50% of the visual tokens, an operating point no baseline we measured reaches on either 7B model. Large video VLMs can therefore be served far more cheaply without giving up accuracy. More broadly, token compression is better posed not as ranking tokens by how important they look, but as asking what a retained visual memory cannot already predict.

1 Introduction

Video vision-language models (VLMs) extend multimodal reasoning from images to temporally rich videos, supporting question answering and reasoning over complex inputs (Xu et al., 2024; Hong et al., 2024; Ren et al., 2024). However, this capability introduces substantial inference cost (Tang et al., 2025). Even with only 16 sampled frames, LLaVA-OneVision produces 3,137 visual tokens, occupying 97.9% of the input context in our profiling setup. This visual-token-dominated context requires 23.8 GB of context-driven memory and accounts for 61.5% of the measured peak memory in LLaVA-OneVision 7B, and the cost is paid again whenever the same video is queried with a different question. These observations make visual token compression essential for practical video VLM deployment, especially in multi-query scenarios such as video chatbots, streaming analysis and meeting assistants (Qian et al., 2024).

For practical deployment, compression benefits from two properties. First, it should be training-free, allowing the method to be applied to existing VLMs without model-specific retraining or additional supervision. Second, it should operate before LLM prefill, since methods that prune after computing LLM attentions have already incurred much of the cost they aim to reduce. A pre-LLM design avoids intrusive language-model modifications and eases deployment across backbones.

Prior work on training-free compression selects visual tokens using saliency, temporal redundancy or attention-based importance (Liu et al., 2025b; Tao et al., 2025; Chen et al., 2024). These criteria work in many settings, but they score a token largely on its own appearance rather than on what it adds to the visual context already retained. A second family instead scores a token against the set already chosen, through feature-space diversity, conditional entropy or facility location (Alvar et al., 2025; Yuan et al., 2026; Cho et al., 2026), which does measure marginal contribution, but in an undirected geometry over the whole token cloud with no temporal reference. A salient token may still be redundant if similar information appears across nearby frames (Fu et al., 2025b). Conversely,

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.

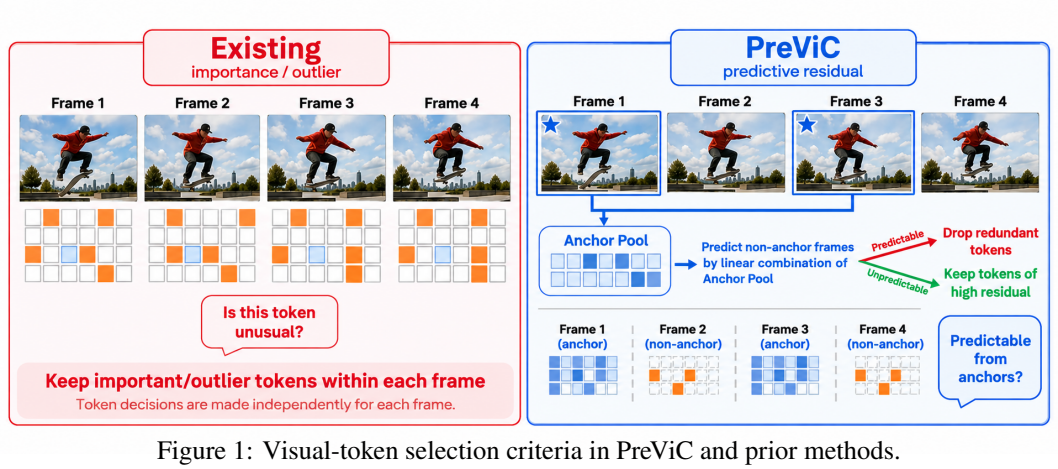

Figure 1: Visual-token selection criteria in PreViC and prior methods.

a subtle temporal change may matter even when it is not a strong outlier, because retained memory cannot predict such changes (Cai et al., 2024; Alexandre et al., 2023).

This limitation motivates us to view video token compression through the lens of predictive coding. Classical video codecs reduce temporal redundancy by maintaining reference frames and encoding only the residual information that cannot be predicted from them (Richardson, 2011; Bross et al., 2021). This principle suggests an analogous criterion for video VLMs. Rather than retaining tokens solely because they appear salient in isolation, compression should prioritize visual information that cannot be explained by a compact memory of the video.

Motivated by this perspective, we propose **PreViC** (**Pre**dictive **Vi**sual **C**ompression), a training-free, prompt-agnostic, pre-LLM method that selects visual tokens according to their predictive residuals with respect to an anchor-based visual memory. As illustrated in Figure 1, existing approaches primarily identify distinctive tokens from the current visual stream, whereas PreViC measures each token relative to previously retained reference information and preserves those that remain difficult to predict. This reframes visual token compression from static importance estimation to reference-aware redundancy modeling.

Concretely, PreViC first selects temporally novel anchor frames to construct a compact visual memory. For each non-anchor token, it retrieves nearby anchor tokens and performs ridge-regularized linear prediction from this reference set. The resulting prediction residual serves as the token importance score. Tokens with large residuals are retained, whereas well-predicted tokens are treated as redundant and removed. A hierarchical budget further preserves anchor information while allocating the remaining capacity according to predictive residuals.

We evaluate PreViC on LLaVA-OneVision (LLaVA-OV) 0.5B and 7B (Li et al., 2024a) and Qwen2.5-VL 7B (Bai et al., 2025) across four video benchmarks against ten baselines. Three checkpoints and five retain ratios give fifteen matched settings. PreViC has the highest four-benchmark average accuracy of any method at 12 of the 15. The one-sided rule of our pre-registered protocol establishes that it matches the uncompressed model within a 1.0 pp margin from $\alpha = 0.2$ on LLaVA-OV 7B, 0.5 on LLaVA-OV 0.5B and 0.7 on Qwen2.5-VL 7B, while no baseline matches at any ratio we measured on either 7B checkpoint. From an efficiency perspective, PreViC also cuts LLaVA-OV 7B LLM prefill time and memory by 70.1% and 68.8%, respectively, at $\alpha = 0.3$. Because PreViC is prompt-agnostic, its compressed visual representation does not depend on the question and can therefore be built once and reused across every later query about the same video, so the cost of producing it is amortized rather than paid again. Overall, these results suggest that video token compression can be framed as reference-aware redundancy modeling before LLM computation.

2 Related Work

Pre-LLM selection. Pre-LLM methods reduce the visual sequence before it enters the language model. DyCoke removes temporally redundant tokens (Tao et al., 2025) and VidCom^2 retains visually distinctive tokens (Liu et al., 2025b). DivPrune instead solves a max-min diversity problem, greedily adding the token whose minimum distance to the already-selected set is largest (Alvar et al., 2025), so its criterion is a set function rather than a per-token score. FastVID combines several

heuristic stages (Shen et al., 2025), ApET ranks tokens by reconstruction error from a compact basis (Ma et al., 2026), and UniComp formulates informational uniqueness through a conditional-entropy objective with frame-group fusion and spatial dynamic compression (Yuan et al., 2026). The last two are the closest to PreViC. Both differ from it in where the reference comes from, and ApET also in where it compresses. ApET's released video path draws its basis geometrically from the token cloud and reconstructs every token of a frame from that one basis, so its references carry no temporal meaning and nothing is ever reconstructed across frames. Dropped tokens are then merged into retained ones rather than discarded. UniComp fuses similar frames into averaged representatives and allocates the budget across frames by a uniqueness score over pooled frame descriptors, so the surviving tokens absorb their neighbors. PreViC keeps whole anchor frames intact as the predictor basis, predicts each remaining token from its own nearest anchors in other frames, and spends the budget on the tokens those anchors fail to predict. DivPrune, UniComp and FLoC (Cho et al., 2026) all score a candidate against the set already chosen, so marginal contribution is not by itself what separates PreViC from prior work. What differs is how a candidate is scored. Their set functions are undirected and atemporal, measuring coverage of the token cloud by whichever tokens happen to be co-selected, whereas PreViC ranks each token by a directed reconstruction residual against an anchor memory, so redundancy is defined by what the anchor frames can already predict.

In-LLM compression. FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2025b), PruneVid (Huang et al., 2025), AIM (Zhong et al., 2025) and HoliTom (Shao et al., 2025) use LLM-layer signals to prune or merge visual tokens. A selector inside the decoder has more information available, since hidden states and attention describe a token more richly than its projected feature does, but it cannot undo the computation already spent on the layers below it and it requires modifying the language model, which makes it harder to deploy across backbones. PruneVid, HoliTom, AIM and ApET act at more than one locus, pruning both before and inside the LLM.

Prompt conditioning. Visual token compression may use the question to identify relevant frames or tokens (Zhang et al., 2025a; Di et al., 2025). This is the easier problem, since only what the question calls for has to be kept, but it forces the video to be compressed again for every new question even when earlier visual features are cached. A question-independent selector can reuse its selected representation but may miss brief, query-specific evidence. FLoC selects query-agnostic representative subsets with a facility-location objective (Cho et al., 2026), and StreamMem keeps a query-agnostic KV cache for streaming video (Yang et al., 2025). PreViC takes this position by design. It compresses without the question so that one representation serves every later query about the video.

Trained and structured alternatives. Learned token compressors train reconstructive or language-aware modules (Liu et al., 2025a; Ye et al., 2025; Wang et al., 2025). AdaCodec is a trained predictive interface with adaptive reference frames (Hou et al., 2026). Approaches of this kind have to train a new compressor for each target VLM, a cost that works against the deployment case compression is meant to serve. PreViC obtains a predictive reference without training, from native post-projector tokens.

3 Method

Problem formulation. A video VLM encodes sampled frames into visual tokens and projects them into the LLM embedding space. We write the projected visual-token tensor as $\mathbf{V} \in \mathbb{R}^{F\times T\times D}$ for F frames, T tokens per frame and hidden dimension D, and write $\mathbf{v}_{t,p} \in \mathbb{R}^D$ for the p-th token of frame t, with $t \in \{0,\dots,F-1\}$ and $p \in \{1,\dots,T\}$. Given a retain ratio $\alpha \in (0,1]$, the implementation clamps it to $\alpha' = \min(1, \max(0.05, \alpha))$ and forms the discrete target $m_0 = \min\{FT, \max[4F, \mathrm{round}(\alpha' FT)]\}$, where the $4F$ term is a four-token-per-frame floor. The returned count m_{real} can depart from m_0 in either direction. The final per-frame top-up raises it, and the non-anchor capacity bound of R3 lowers it on short inputs, so the implementation records m_{real} rather than inferring it from α. Selected tokens enter the LLM in temporal order. PreViC is training-free and prompt-agnostic. Its keep mask $\mathbf{M} = f(\mathbf{V}) \in \{0,1\}^{F\times T}$, whose entries sum to m_{real}, depends only on visual tokens, so the compressed feature can be cached and reused across questions about the same video.

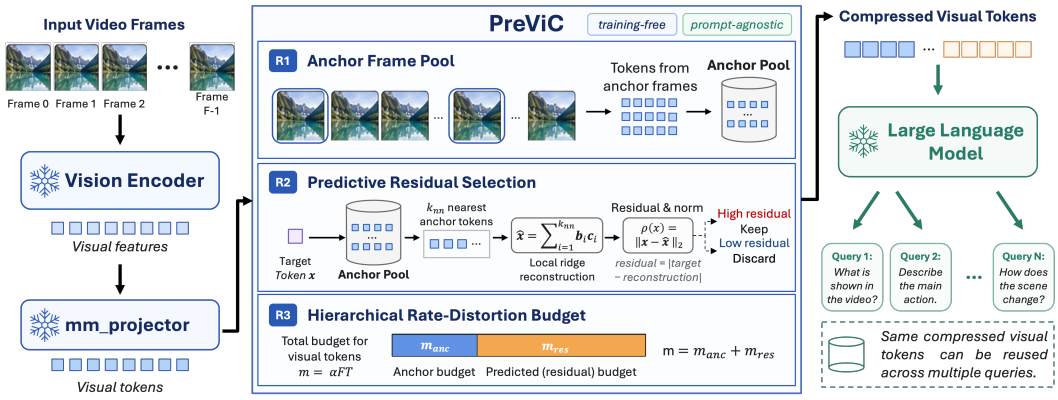

Figure 2: Overview of PreViC's R1, R2 and R3 visual-token selection pipeline.

Overview. Figure 2 summarizes the PreViC pipeline. PreViC adopts a predictive-coding view. Rather than scoring tokens by static saliency or diversity, it asks whether each token can be predicted from a compact visual memory. R1 constructs an anchor-frame pool, R2 scores non-anchor tokens by predictive residuals, and R3 allocates the budget between anchor and high-residual non-anchor tokens. R2 is the main selection rule. R1 and R3 supply the reference structure and an anchor-preservation floor. Algorithm 1 in Appendix A gives the pseudocode of PreViC.

R1: anchor-frame pool. To build a compact visual memory, PreViC scores each frame $t > 0$ by its frame-level temporal novelty ν_t, the average token-wise change from the previous frame:

$$\nu_t = \frac{1}{T} \sum_{p=1}^{T} \left\| \mathbf{v}_{t,p} - \mathbf{v}_{t-1,p} \right\|_2^2 \tag{1}$$

This score is a lightweight feature-space analogue of codec-style scene-cut detection (Sastre et al., 2006). Frames with high novelty are less predictable from preceding frames and therefore serve as useful anchors for later residual prediction. We set $\nu_0 = +\infty$ so the first frame is always an initial reference, take the top-a frames by ν_t as the anchor indices $\mathcal{A} = \mathrm{Top}_a(\nu_0, \ldots, \nu_{F-1})$, and define the anchor pool as all of their tokens, $\mathcal{P} = \{\mathbf{v}_{t,p} \mid t \in \mathcal{A},\ p = 1, \ldots, T\}$. The pool is the visual memory, analogous to reference frames in predictive video coding.

R2: predictive residual scoring. For each non-anchor token $\mathbf{x} = \mathbf{v}_{t,p}$ with $t \notin \mathcal{A}$, PreViC retrieves from $\mathcal{P}$ the k_{nn} nearest anchor tokens by cosine similarity, written $\mathcal{N}_{k_{\mathrm{nn}}}(\mathbf{x})$, and stacks them row-wise into the local anchor basis $\mathbf{B}_\mathbf{x} = [\mathbf{b}_1^\top; \ldots; \mathbf{b}_{k_{\mathrm{nn}}}^\top] \in \mathbb{R}^{k_{\mathrm{nn}} \times D}$, then predicts $\mathbf{x}$ from that basis by ridge regression:

$$\begin{aligned} \mathbf{c}_\mathbf{x}^* &= \arg\min_{\mathbf{c}_\mathbf{x} \in \mathbb{R}^{k_{\mathrm{nn}}}} \left\| \mathbf{B}_\mathbf{x}^\top \mathbf{c}_\mathbf{x} - \mathbf{x} \right\|_2^2 + \lambda \left\| \mathbf{c}_\mathbf{x} \right\|_2^2 \\ &= \left(\mathbf{B}_\mathbf{x} \mathbf{B}_\mathbf{x}^\top + \lambda \mathbf{I} \right)^{-1} \mathbf{B}_\mathbf{x} \mathbf{x}. \end{aligned} \tag{2}$$

The predicted token is $\hat{\mathbf{x}} = \mathbf{B}_\mathbf{x}^\top \mathbf{c}_\mathbf{x}^*$ and the predictive residual score is $\rho(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2$. A large residual means $\mathbf{x}$ carries information the anchor memory does not explain, and a small one means it is predictable from existing anchors (the squared norm gives the same ranking). Neighbors are retrieved by cosine similarity for scale insensitivity, the residual is measured on raw post-projector features, and λ is set adaptively for numerical stability. Because each token is fitted only from its own k_{nn} nearest anchors, Equation 2 assumes predictability within a local neighborhood rather than global linearity of the representation. Appendix D gives Equation 4 and the scoring-space details.

R3: hierarchical rate allocation. R3 allocates the discrete pre-top-up target m_0 between anchor tokens and residual-selected non-anchor tokens. Anchor tokens form the reference memory and are not scored by R2, so a ranking of non-anchor residuals alone would omit them. R3 therefore reserves a nominal anchor budget $\widehat{m}_{\mathrm{anc}} = \mathrm{round}(\beta m_0)$, capped at the anchor-pool size $N_{\mathrm{anc}} = aT$ so that $m_{\mathrm{anc}} = \min(\widehat{m}_{\mathrm{anc}}, N_{\mathrm{anc}})$, leaving $m_{\mathrm{res}} = m_0 - m_{\mathrm{anc}}$ for the $N_{\mathrm{res}} = (F - a)T$ non-anchor tokens. Thus β is a target anchor fraction, not a guaranteed final one. Every anchor token is kept when the cap binds. Otherwise each anchor token $\mathbf{q} \in \mathcal{P}$ is ranked by its distance $s(\mathbf{q}) = \|\mathbf{q} - \bar{\mathbf{q}}\|_2$

to the anchor-pool centroid $\bar{\mathbf{q}}$ and the top m_{anc} are kept. PreViC then retains the top $\min[m_{\text{res}}, N_{\text{res}}]$ non-anchor tokens by $\rho(\mathbf{x})$, and any frame left with fewer than four tokens is topped up from its first unselected positions. R3 is an anchor-preservation floor rather than the primary ranking criterion. Residuals are computed against the full anchor pool $\mathcal{P}$, while R3 decides which anchor tokens are emitted.

4 Experiments

4.1 Experimental Setup

Models and benchmarks. We evaluate PreViC and the baselines on LLaVA-OneVision 0.5B and 7B and on Qwen2.5-VL 7B, and report MVBench (Li et al., 2024b), Video-MME (Fu et al., 2025a), EgoSchema (Mangalam et al., 2023) and LongVideoBench (Wu et al., 2024). The primary metric is the four-benchmark average, the unweighted mean of the four scores.

Evaluation protocol. Every score path uses the official `lmms-eval` checkout, greedy decoding at batch size 1 and at most 32 sampled frames. Benchmark populations are never pooled, and every accuracy interval comes from a paired cluster bootstrap that resamples each benchmark's declared video or task–video cluster rather than individual questions, so a video's questions move together.

Baselines. We compare PreViC with DivPrune (Alvar et al., 2025), DyCoke (Tao et al., 2025), FastVID (Shen et al., 2025), PruneVid (Huang et al., 2025), AIM (Zhong et al., 2025), HoliTom (Shao et al., 2025), FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2025b), ApET (Ma et al., 2026) and UniComp (Yuan et al., 2026). Table 1 records where each method removes tokens and what guides the selection. All of them select on visual information, and the set spans compression before the LLM, inside it and at both loci, under visual-only and question-conditioned guidance.

Table 1: Taxonomy of the baselines by compression locus and use of text guidance.

Method	Venue	Pre-LLM	In-LLM	KV-cache	Text info
FastV	ECCV '24		✓		✓
DivPrune	CVPR '25	✓			
DyCoke	CVPR '25	✓		✓	✓
AIM	ICCV '25	✓	✓		✓
SparseVLM	ICML '25		✓		✓
PruneVid	ACL '25 Findings	✓	✓		✓
FastVID	NeurIPS '25	✓			
HoliTom	NeurIPS '25	✓	✓		✓
ApET	CVPR '26	✓	✓		
UniComp	CVPR '26	✓			
PreViC (Ours)	–	✓			

4.2 Accuracy at Matched Retain Ratio

Table 2 reports four-benchmark average accuracy for PreViC and the ten baselines at the five retain ratios (0.1, 0.2, 0.3, 0.5, 0.7), on all three checkpoints, with Vanilla as the uncompressed reference. Figure 3 plots the same quantity against the retain ratio, with the uncompressed model as a dashed line and a shaded band 1.0 pp below it. Both displays rank methods at a matched requested ratio.

Every cell of Table 2 averages four benchmark scores, so the fifteen checkpoint and ratio settings rest on 60 evaluation runs per method. PreViC has the highest average of any method at **12 of the 15** settings, five of five on LLaVA-OV 7B, four of five on Qwen2.5-VL 7B and three of five on LLaVA-OV 0.5B. Against the strongest baseline the paired 90% interval lies above zero at six of these, all on the 7B checkpoints (Appendix B.5). Table 8 in Appendix B.2 collapses each baseline to one number per checkpoint, the amount by which PreViC leads that baseline on average across the five shared retain ratios, and its last column averages that lead over the three checkpoints. Read this way, PreViC leads every baseline, by 1.12 pp against the closest of them, PruneVid, and by 6.18 pp against the furthest, SparseVLM. Table 9 in Appendix B.2 breaks the average into its four benchmarks at $\alpha \in \{0.2, 0.5, 0.7\}$ for PreViC and seven of the baselines.

At the three remaining settings one baseline edges ahead, and two of them fall where the suite resolves nothing. PruneVid leads by 0.02 pp at $\alpha = 0.2$ and DivPrune by 0.15 pp at 0.5 on LLaVA-OV 0.5B, where PreViC itself gives up only 0.34 pp even at the sparsest budget we measure. Both leads are below the suite's worst-case standard error of the average, 0.7255 pp. The third is PruneVid by 0.85 pp at $\alpha = 0.1$ on Qwen2.5-VL 7B, the sparsest budget on the hardest checkpoint. Section 4.3 shows that none of the three is a difference this evidence establishes.

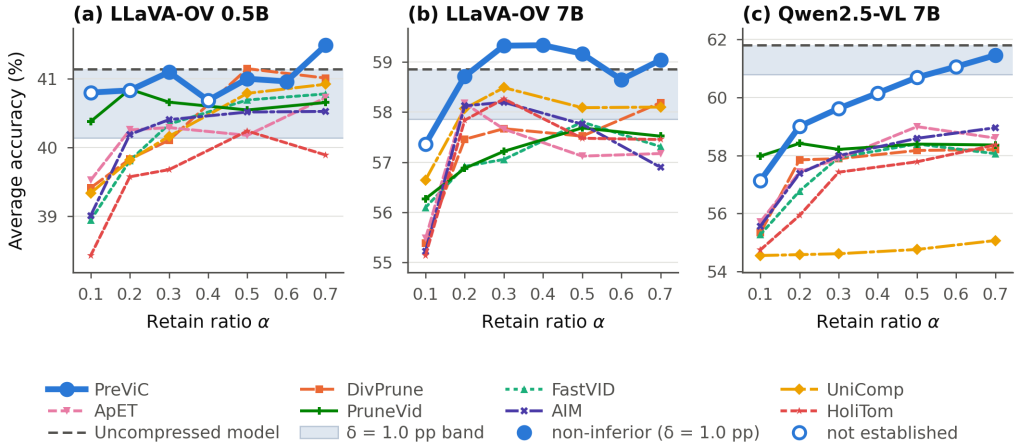

Figure 3: Four-benchmark average accuracy (%) against the retain ratio α, one panel per checkpoint, for PreViC at seven retain ratios and seven of the ten baselines at five (all ten are in Appendix B.2). The dashed line marks the uncompressed model and the shaded band the 1.0 pp margin below it. Filled PreViC markers are the ones the one-sided rule establishes, which is the weaker of the two verdicts and not the two-sided count of Table 3.

Figure 3 shows the same comparison as a function of the budget. PreViC tracks the top of the fan across the grid. Its lead over whichever baseline is best at each individual setting averages 0.23 pp on LLaVA-OV 0.5B, 0.82 pp on LLaVA-OV 7B and 1.07 pp on Qwen2.5-VL 7B, so the margin is widest on the checkpoints where compression costs the most. The three panels differ in how much room there is for a margin at all. On LLaVA-OV 0.5B the curves are flat and crowded against the reference, and every plotted curve stays within three points of it at every budget. Qwen2.5-VL 7B is the opposite case, where compression is expensive and the curves separate. At $\alpha = 0.7$ on that checkpoint, PreViC is 0.34 pp short of its reference while the best baseline at that budget is still 2.84 pp below it. LLaVA-OV 7B sits between the two. A matched-ratio ranking is therefore informative in proportion to how much the checkpoint loses under compression, and the selection rule matters most where compression is expensive.

4.3 Matching the Uncompressed Model

Table 3 counts how often each method's four-benchmark average matches the uncompressed model's within a 1.0 pp margin. A coordinate matches when the interval on the paired difference, not the point estimate alone, lies inside that margin. The bootstrap, the 90% interval and the margin were

Table 2: Four-benchmark average accuracy (%) at matched retain ratio α, for PreViC and all ten baselines (unweighted mean of MVBench, Video-MME, EgoSchema and LongVideoBench). Bold marks the highest compressed value, decided before rounding.

	LLaVA-OV 0.5B					LLaVA-OV 7B					Qwen2.5-VL 7B				
Method	.10	.20	.30	.50	.70	.10	.20	.30	.50	.70	.10	.20	.30	.50	.70
Vanilla			41.1					58.9					61.8		
FastV	37.2	38.8	39.6	39.9	40.3	52.7	55.3	56.4	57.2	57.2	53.8	56.4	57.5	58.3	58.8
DivPrune	39.4	39.8	40.1	**41.1**	41.0	55.4	57.5	57.7	57.5	58.2	55.3	57.8	57.9	58.2	58.2
DyCoke	40.1	40.1	40.7	40.6	40.8	56.4	56.4	56.3	57.3	57.9	55.2	55.2	55.2	58.3	58.8
AIM	39.0	40.2	40.4	40.5	40.5	55.2	58.1	58.2	57.8	56.9	55.6	57.4	58.0	58.6	59.0
SparseVLM	35.7	35.8	36.0	35.9	35.8	52.3	52.3	52.3	52.4	52.3	52.6	52.5	52.8	52.7	52.6
PruneVid	40.4	**40.8**	40.7	40.5	40.7	56.3	56.9	57.2	57.7	57.5	**58.0**	58.4	58.2	58.4	58.4
FastVID	38.9	39.8	40.3	40.7	40.8	56.1	56.9	57.1	57.8	57.3	55.3	56.8	57.9	58.4	58.1
HoliTom	38.4	39.6	39.7	40.2	39.9	55.1	57.8	58.3	57.5	57.5	54.7	55.9	57.4	57.8	58.4
ApET	39.5	40.3	40.3	40.2	40.7	55.5	58.2	57.7	57.1	57.2	55.7	57.4	57.9	59.0	58.6
UniComp	39.3	39.8	40.2	40.8	40.9	56.6	58.1	58.5	58.1	58.1	54.5	54.6	54.6	54.8	55.1
PreViC	**40.8**	40.8	**41.1**	41.0	**41.5**	**57.4**	**58.7**	**59.4**	**59.2**	**59.0**	57.1	**59.0**	**59.6**	**60.7**	**61.5**

Table 3: Two-sided matches with the uncompressed model within the 1.0 pp margin, counted over the five retain ratios all methods share, for PreViC and all ten baselines (on four-benchmark average accuracy). The final row totals the ten baselines.

Method	LLaVA-OV 0.5B	LLaVA-OV 7B	Qwen2.5-VL 7B	Total
PreViC	3 / 5	3 / 5	1 / 5	7 / 15
DivPrune	2 / 5	0 / 5	0 / 5	2 / 15
FastVID	2 / 5	0 / 5	0 / 5	2 / 15
UniComp	1 / 5	0 / 5	0 / 5	1 / 15
ApET	1 / 5	0 / 5	0 / 5	1 / 15
PruneVid	1 / 5	0 / 5	0 / 5	1 / 15
AIM	0 / 5	0 / 5	0 / 5	0 / 15
HoliTom	0 / 5	0 / 5	0 / 5	0 / 15
DyCoke	1 / 5	0 / 5	0 / 5	1 / 15
FastV	0 / 5	0 / 5	0 / 5	0 / 15
SparseVLM	0 / 5	0 / 5	0 / 5	0 / 15
All baselines (10)	8	0	0	8

pre-registered before any result under them was read (Appendix B.1), with the margin sitting above the suite's worst-case standard error of 0.7255 pp. Appendix B.5 carries every coordinate this protocol governs. Table 16 lists all 24 PreViC-versus-uncompressed budgets with their intervals, and Tables 17–19 do the same for the ten baselines, one table per checkpoint.

PreViC matches at 7 of its own 15 coordinates, one coordinate for each of the three checkpoints at each of the five shared retain ratios. A coordinate counts as a match when that interval lies inside ± 1.0 pp, so what the count reports is how often the evidence places the difference within the margin, not how often a difference went undetected. The one-sided rule below is the weaker of the two and asks only that the interval stay above -1.0 pp. The ten baselines reach 8 between them, all on LLaVA-OV 0.5B and none on either 7B checkpoint. They clear the bar only on the checkpoint where compression costs least, the one on which PreViC itself loses only 0.34 pp at 90% discard. Under the one-sided rule PreViC matches from $\alpha = 0.2$ upward on LLaVA-OV 7B, from 0.5 on LLaVA-OV 0.5B and from 0.7 on Qwen2.5-VL 7B. The sweep is not monotone, since $\alpha = 0.4$ on LLaVA-OV 0.5B is not established while 0.3 is, so we claim thresholds rather than budgets.

The two 7B checkpoints are where the methods separate, and neither admits a single baseline. On LLaVA-OV 7B none of the 50 baseline coordinates is established within the margin, the closest being UniComp at $\alpha = 0.3$, $\Delta = -0.371$ pp, $[-1.237, +0.495]$, whose interval still reaches past the margin. PreViC matches there from $\alpha = 0.2$ upward, at seven of the eight budgets we measured. On Qwen2.5-VL 7B the gap to the reference is far larger for every method. PreViC stays below the uncompressed 61.80% reference at every measured budget, by 4.67 pp at $\alpha = 0.1$ and 1.11 pp at 0.5, and is first established within the margin at 0.7. None of the 50 Qwen baseline coordinates is established within the margin either, the closest being ApET at $\alpha = 0.5$, $\Delta = -2.806$ pp, $[-3.778, -1.817]$, entirely below the margin. PreViC is therefore the only method that matches either 7B reference within the margin at any measured budget. The same estimator applied to the PreViC-versus-baseline differences separates no baseline from PreViC in its own favor at any of the 150 coordinates, including the three settings where a baseline's average is higher (Appendix B.5). Where those intervals overlap, the ordering is not one this evidence resolves.

4.4 Accuracy and Efficiency at an Aggressive Operating Point

Table 4 pairs full-population LLaVA-OV 7B accuracy at $\alpha = 0.3$ with single-input time-to-first-token (TTFT) for PreViC and the baselines. This is an operating-point diagnostic alongside the matched-ratio comparison of Table 2. At this budget PreViC answers in 3.131 s against the uncompressed model's 3.823 s, a reduction of 18.1% measured on the same GPU in the same job. Table 5 decomposes the Vanilla and PreViC rows of that table into the stages, and compression reaches one of them. LLM prefill falls from 973.9 to 291.0 ms, and the memory it adds above what is already resident falls from 5.78 to 1.80 GiB, reductions of 70.1% and 68.8%, respectively. These results are measured on an NVIDIA RTX 6000 Ada Generation GPU.

Table 4: LLaVA-OV 7B average accuracy (%) and TTFT (s).

Method	TTFT (s)	Avg.
Vanilla	3.823	58.86
DivPrune	3.361	57.67
DyCoke	3.175	56.31
FastVID	3.197	57.05
PruneVid	4.590	57.22
AIM	3.244	58.19
HoliTom	3.267	58.26
FastV	3.319	56.45
SparseVLM	3.145	52.33
ApET	3.259	57.66
UniComp	3.318	58.49
PreViC	3.131	**59.38**

Table 5: Stage decomposition of the LLaVA-OV 7B time to first token (ms), with the memory the prefill stage allocates. The five stage rows sum to TTFT.

Stage	Vanilla	α=0.3	α=0.5
Video decode + preproc.	2,427.3	2,403.2	2,425.0
Vision tower	415.95	416.48	417.13
Projector	5.57	5.65	5.70
Compressor	–	14.43	15.12
Prefill	973.93	290.97	448.00
TTFT	**3,822.7**	**3,130.8**	**3,311.0**
Decode (15 forwards)	443.96	421.08	436.83
Prefill mem. (GiB)	5.78	1.80	2.96
Prefill time saving	–	70.1%	54.0%
Prefill mem. saving	–	68.8%	48.9%

4.5 Same-Video Multi-Query Prefix Reuse

Figure 4 plots cumulative first-token LLM time against the number of questions asked about a single video, for a path that rebuilds the visual prefix for every question and for one that builds it once and reads it back. Because PreViC never sees the question, the compressed prefix is the same for every question about that video, so its key and value tensors need only be built once, and the figure asks what that is worth as questions accumulate. This reuse is an advantage of prompt-agnostic compression, which PreViC adopts by design (Section 5). The two curves separate from the left and keep separating. Recomputation grows in proportion to the number of questions while reuse grows far more slowly, each further question reading a prefix that already exists, and by the right-hand end of the plot the gap exceeds an order of magnitude. Table 21 in Appendix B.6 gives the plotted values.

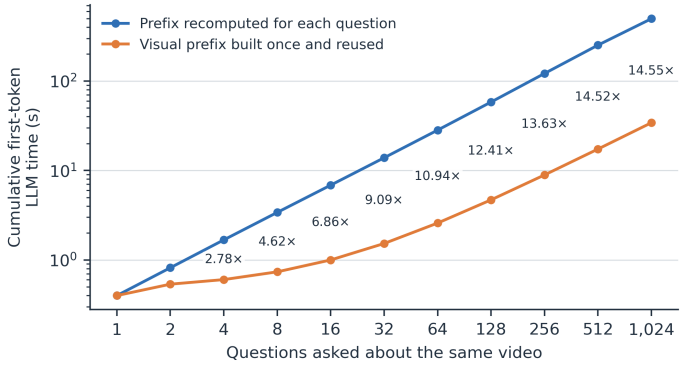

Figure 4: Cumulative first-token LLM time against the number of questions asked about one video.

4.6 Component Ablation

Table 6 reports LLaVA-OV 7B accuracy on MVBench at $\alpha = 0.1$, 0.2 and 0.3 for full PreViC and two controls that each replace one selection rule with a random choice. The sparsest budgets are where selection matters most, and MVBench is our largest population. Ranking by predictive residual matters most. Replacing it with a random order costs 2.225, 1.475 and 0.900 pp as the budget widens, the largest cost at every budget. Choosing the reference by temporal novelty rather than at random adds 0.950, 0.450 and 0.050 pp, a difference no longer measurable by $\alpha = 0.3$. Both costs shrink because a wider budget leaves less for any rule to decide, and full PreViC is ahead of both controls at every budget. The residual ranking is therefore what separates PreViC from its controls, with the anchor memory supplying the reference it is measured against.

Table 6: Component ablation, LLaVA-OV 7B on MVBench. w/o R1 predicts from $2T$ random video tokens instead of the anchor frames, and w/o R2 ranks non-anchor tokens at random.

			MVBench accuracy (%)		
Variant	R1	R2	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$
Full PreViC	✓	✓	57.800	58.200	58.150
w/o R1	–	✓	56.850	57.750	58.100
w/o R2	✓	–	55.575	56.725	57.250

5 Discussion

Predictive residuals for query-independent compression. PreViC measures visual redundancy relative to an anchor memory rather than question-specific relevance. This prompt-agnostic design is deliberate, letting one compressed representation serve every later query (Section 4.5). Reconstruction difficulty, however, may reflect both semantic changes and low-level visual variations such as camera motion or scene transitions. We therefore interpret the residual as a reference-conditioned predictability signal, and our results support its use for token selection at the evaluated budgets.

Latency and memory implications of visual token compression. The potential latency benefit of visual token compression is constrained by the fraction of computation affected by token reduction. On LLaVA-OV 7B, LLM prefill accounts for 25.5% of TTFT (Table 5). With the remaining stages unchanged, this fraction imposes an upper bound on the TTFT reduction achievable through prefill acceleration alone. Memory savings are not subject to the same bound. Because prefill processes the retained token sequence jointly, its memory allocation can decrease more directly with the number of retained tokens. At $\alpha = 0.3$, prefill memory allocation decreases from 5.78 to 1.80 GiB, a 68.8% reduction that approximately follows the reduction in token count. These results indicate that visual token compression provides a more pronounced benefit in memory consumption than in end-to-end latency, making it particularly valuable for memory-constrained deployment.

Task-dependent limitations of redundancy-based selection. Prioritizing unpredictable information can suppress repeated evidence that remains necessary for downstream tasks. The per-task results illustrate this trade-off (Table 13). PreViC improves performance on action prediction, counterfactual inference, state change and action sequence, which emphasize events, transitions and temporal relationships. In contrast, moving count and fine-grained action are the only tasks with consistent performance decreases across all three checkpoints, with moving count exhibiting the largest individual degradation among the evaluated tasks. This pattern is consistent with a limitation of treating predictability as a proxy for dispensability. For example, when multiple objects exhibit similar motion, tokens representing distinct instances may nevertheless be predictable from the anchor memory. Discarding such tokens can remove evidence needed for accurate counting. More generally, tasks requiring exhaustive enumeration or fine-grained discrimination may depend on information that is predictable yet task-relevant, limiting the suitability of redundancy-based selection in these settings.

6 Conclusion

We presented PreViC, a training-free and prompt-agnostic pre-LLM compression method for video VLMs. Instead of estimating token importance in isolation, PreViC retains visual tokens that a compact memory of temporally novel anchor frames cannot predict, reframing compression as reference-aware redundancy modeling. Across fifteen checkpoint and retain-ratio settings, it has the highest four-benchmark average accuracy of any method at 12 of them, and it matches the uncompressed model on both 7B checkpoints at budgets where no measured baseline does. PreViC attains this from the least informed position in the design space, observing only the projected visual sequence while seven of the ten baselines additionally exploit signals from within the language model, the question or the key–value cache. That the richer signal did not decide these comparisons supports a broader principle for efficient video VLMs, namely that visual tokens are more usefully scored by what they contribute beyond a retained visual memory than by how salient they appear in isolation.

AI Usage Statement

Generative AI tools were used in four ways in this work. They helped search and organize the related literature, assisted in writing implementation and test code, performed grammar and copy-editing passes over the manuscript, and produced internal reviews of the draft that the authors then acted on. Every reported number comes from the recorded evaluation runs rather than from generated text, including benchmark predictions, accuracy scores, measured timings and the statistics computed from them. The authors verified AI-assisted work against the primary literature and the recorded artifacts, and take full responsibility for the content of this paper.

Ethics Statement

This work evaluates publicly available model checkpoints, benchmark datasets and baseline implementations under their respective licenses and evaluation terms. It collects no new human-subject data and redistributes no videos, annotations or model weights. Because compression changes which visual evidence reaches the language model, a compressed deployment can fail on evidence that the uncompressed model retains, and the paper reports the settings where this happens rather than presenting compression as uniformly beneficial.

Reproducibility Statement

Section 4.1 states the evaluator, the benchmarks and the checkpoints under which every method is compared, and Appendix B.1 states the evaluation protocol, which was pre-registered and hashed before any result under it was read. Appendix A gives the notation and the pseudocode for the selector. An anonymized implementation is available at `https://anonymous.4open.science/r/previc-86F1/`.

References

David Alexandre, Hsueh-Ming Hang, and Wen-Hsiao Peng. Hierarchical b-frame video coding using two-layer canf without motion coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10249–10258, 2023.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, et al. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Proceedings of the Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

Benjamin Bross, Jianle Chen, Jens-Rainer Ohm, Gary J Sullivan, and Ye-Kui Wang. Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc). *Proceedings of the IEEE*, 109(9):1463–1493, 2021.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

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 *Proceedings of the European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Janghoon Cho, Jungsoo Lee, Munawar Hayat, Kyuwoong Hwang, Fatih Porikli, and Sungha Choi. Floc: Facility location-based efficient visual token compression for

long video understanding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a87d3a6b4884108c0dff392c1c7e2d82-Abstract-Conference.html.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. Streaming video question-answering with in-context video kv-cache retrieval. In *Proceedings of the International Conference on Learning Representations*, volume 2025, pp. 42115–42127, 2025.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Framefusion: Combining similarity and importance for video token reduction on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22654–22663, 2025b.

Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.

Haowen Hou, Zhen Huang, Zheming Liang, Qingyi Si, Chenglin Li, Shuai Dong, Kele Shao, Ruilin Li, Dianyi Wang, Nan Duan, and Jiaqi Wang. Adacodec: A predictive visual code for video mllms. *arXiv preprint arXiv:2606.02569*, 2026.

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

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

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, 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.

Xiangrui Liu, Yan Shu, Zheng Liu, Ao Li, Yang Tian, and Bo Zhao. Video-xl-pro: Reconstructive token compression for extremely long video understanding. *arXiv preprint arXiv:2503.18478*, 2025a.

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, 2025b.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. Apet: Approximation-error guided token compression for efficient vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26306–26316, 2026.

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

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Proceedings of the Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14313–14323, 2024.

Iain E Richardson. *The H.264 Advanced Video Compression Standard*. John Wiley & Sons, 2011.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Jorge Sastre, Pau Usach, A Moya, Valery Naranjo, and JM López. Shot detection method for low bitrate H.264 video coding. In *2006 14th European Signal Processing Conference*, pp. 1–5. IEEE, 2006.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. *Proceedings of the Advances in Neural Information Processing Systems*, 38:135547–135570, 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. *Proceedings of the Advances in Neural Information Processing Systems*, 38:123553–123581, 2025.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

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

Xiangchen Wang, Jinrui Zhang, Teng Wang, Haigang Zhang, and Feng Zheng. Seeing more, saying more: Lightweight language experts are dynamic video token compressors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 541–558, 2025.

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

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

Yanlai Yang, Zhuokai Zhao, Satya Narayan Shukla, Aashu Singh, Shlok Kumar Mishra, Lizhu Zhang, and Mengye Ren. Streammem: Query-agnostic kv cache memory for streaming video understanding. *arXiv preprint arXiv:2508.15717*, 2025.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29836–29846, 2025.

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

Shaojie Zhang, Jiahui Yang, Jianqin Yin, Zhenbo Luo, and Jian Luan. Q-frame: Query-aware frame selection and multi-resolution adaptation for video-llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22056–22065, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning*, pp. 74840–74857. PMLR, 2025b.

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

A Notation and Pseudocode

We use lowercase letters for scalars, bold lowercase letters for vectors, bold uppercase letters for matrices or tensors and calligraphic letters for sets. Table 7 lists every symbol the selector uses. The statistical symbols of Sections 4.1–5 (Δ, δ, B and n) are defined in Appendix B.1 and Table 16, and p always denotes the within-frame token index. Retain ratio means the requested setting throughout (Section 3), and it is the budget axis of every comparison in the paper.

Three of the symbols are specific to this implementation and have no counterpart in a formulation that writes the budget simply as $\lceil \alpha FT \rceil$. They are the clamped ratio α', the discrete pre-top-up target m_0 and the returned count m_{real}. We keep them because the selector does not compute a ceiling. It clamps α to $[0.05, 1]$, rounds ties to even, imposes a four-token-per-frame floor and an FT ceiling, and then returns a count that the per-frame top-up can raise above m_0 and the non-anchor capacity bound N_{res} can leave below it. Both the floor and the gap between target and returned tokens are discussed in the paper, so collapsing the three symbols into one budget symbol would misdescribe the selector.

Algorithm 1 gives the pseudocode for PreViC's visual token compression procedure.

Table 7: Notation for the fixed PreViC selector.

Symbol	Meaning	Symbol	Meaning
$\mathbf{V} \in \mathbb{R}^{F\times T\times D}$	Projected visual-token tensor after the multimodal projector	$\mathbf{v}_{t,p} \in \mathbb{R}^D$	p-th projected visual token in frame t
F	Number of sampled video frames	T	Number of visual tokens per frame
D	Hidden dimension of each projected visual token	t	Frame index, $t \in \{0, \dots, F-1\}$
p	Token index within a frame, $p \in \{1, \dots, T\}$	$\alpha \in (0, 1]$	Retain ratio, by definition the requested setting
α'	Implementation-clamped retain ratio $\min(1, \max(0.05, \alpha))$	round	Rounding to the nearest integer, with Python's ties-to-even rule
m_0	Discrete pre-top-up target $\min\{FT, \max[4F, \text{round}(\alpha' FT)]\}$	m_{real}	Returned visual-token count, after the per-frame floor and the N_{res} bound
$\mathbf{M} \in \{0,1\}^{F\times T}$	Binary keep mask whose entries sum to m_{real}	$\tilde{\mathbf{V}}$	Compressed output visual-token sequence
a	Number of anchor frames (default 2)	$\mathcal{A}$	Set of selected anchor-frame indices
ν_t	Frame-level temporal novelty score for frame t	$\mathcal{P}$	Anchor-token pool built from the frames in $\mathcal{A}$
$\mathbf{x}$	Non-anchor target token to be scored	k_{nn}	Number of nearest anchor tokens used for local reconstruction (default 4)
$\mathcal{N}_{k_{\text{nn}}}(\mathbf{x})$	Set of k_{nn} nearest anchor tokens for $\mathbf{x}$	$\mathbf{B}_{\mathbf{x}} \in \mathbb{R}^{k_{\text{nn}}\times D}$	Local anchor basis retrieved for $\mathbf{x}$
$\mathbf{b}_i \in \mathbb{R}^D$	i-th anchor token in the local basis $\mathbf{B}_{\mathbf{x}}$	$\mathbf{c}_{\mathbf{x}} \in \mathbb{R}^{k_{\text{nn}}}$	Reconstruction coefficient vector for $\mathbf{x}$
$\hat{\mathbf{x}} \in \mathbb{R}^D$	Reconstructed token $\mathbf{B}_{\mathbf{x}}^\top \mathbf{c}_{\mathbf{x}}^*$	$\rho(\mathbf{x})$	Predictive residual score of token $\mathbf{x}$
λ	Ridge regularization coefficient, Equation 4	ϵ	Ridge regularization scale (10^{-3} by default)
$\mathbf{I}$	Identity matrix	$\beta \in [0, 1]$	Anchor budget fraction (default 0.3)
$\widehat{m}_{\text{anc}}$	Nominal anchor budget $\text{round}(\beta m_0)$, before capping	m_{anc}	Final anchor budget $\min(\widehat{m}_{\text{anc}}, N_{\text{anc}})$
$m_{\text{res}} = m_0 - m_{\text{anc}}$	Budget for residual-scored non-anchor tokens	$N_{\text{anc}} = aT$	Anchor-pool size (number of anchor tokens)
$N_{\text{res}} = (F-a)T$	Number of non-anchor tokens, which caps m_{res} without redistribution	$\mathbf{q} \in \mathbb{R}^D$	Anchor token, a generic element of the anchor pool $\mathcal{P}$
$\bar{\mathbf{q}} \in \mathbb{R}^D$	Centroid of the anchor pool $\mathcal{P}$	$s(\mathbf{q})$	Anchor-token score used when $\lvert\mathcal{P}\rvert$ exceeds m_{anc}
$\mathcal{S}_{\text{anc}}$	Selected anchor-token set	$\mathcal{S}_{\text{res}}$	Selected residual-token set

Algorithm 1 PreViC

Require: Visual tokens $\mathbf{V} \in \mathbb{R}^{F\times T\times D}$, retain ratio α, anchor count a, anchor budget fraction β, neighbor count k_{nn}, ridge scale ϵ
Ensure: Compressed visual token sequence $\tilde{\mathbf{V}}$

$\alpha' \leftarrow \min(1, \max(0.05, \alpha))$
$m_0 \leftarrow \min\{FT, \max[4F, \text{round}(\alpha' FT)]\}$
Compute frame novelty $\nu_t = \frac{1}{T}\sum_{p=1}^{T} \|\mathbf{v}_{t,p} - \mathbf{v}_{t-1,p}\|_2^2$ for $t > 0$
Set $\nu_0 \leftarrow +\infty$
$\mathcal{A} \leftarrow \text{Top}_a(\nu_0, \ldots, \nu_{F-1})$
$\mathcal{P} \leftarrow \{\mathbf{v}_{t,p} \mid t \in \mathcal{A},\ p = 1, \ldots, T\}$
$N_{\text{anc}} \leftarrow |\mathcal{P}| = aT;\ N_{\text{res}} \leftarrow (F-a)T$
$\widehat{m}_{\text{anc}} \leftarrow \text{round}(\beta m_0)$
$m_{\text{anc}} \leftarrow \min(\widehat{m}_{\text{anc}}, N_{\text{anc}})$
$m_{\text{res}} \leftarrow m_0 - m_{\text{anc}}$
if $m_{\text{anc}} = N_{\text{anc}}$ **then**
    $\mathcal{S}_{\text{anc}} \leftarrow \mathcal{P}$
else
    $\bar{\mathbf{q}} \leftarrow \frac{1}{N_{\text{anc}}}\sum_{\mathbf{q}\in\mathcal{P}} \mathbf{q}$
    Score each anchor token by $s(\mathbf{q}) = \|\mathbf{q} - \bar{\mathbf{q}}\|_2$
    $\mathcal{S}_{\text{anc}} \leftarrow \text{Top}_{m_{\text{anc}}}\{s(\mathbf{q}) \mid \mathbf{q} \in \mathcal{P}\}$
end if
for all $\mathbf{x} = \mathbf{v}_{t,p}$ where $t \notin \mathcal{A}$ **do**
    $\mathcal{N}_{k_{\text{nn}}}(\mathbf{x}) \leftarrow \text{Top}_{k_{\text{nn}}}\{\cos(\mathbf{x}, \mathbf{q}) \mid \mathbf{q} \in \mathcal{P}\}$
    Form $\mathbf{B}_{\mathbf{x}} = [\mathbf{b}_1^\top; \ldots; \mathbf{b}_{k_{\text{nn}}}^\top]$ from $\mathcal{N}_{k_{\text{nn}}}(\mathbf{x})$
    $\lambda \leftarrow \max\big(\epsilon,\ \epsilon \cdot \text{tr}(\mathbf{B}_{\mathbf{x}}\mathbf{B}_{\mathbf{x}}^\top)/k_{\text{nn}}\big)$ ▷ $\epsilon = 10^{-3}$, Equation 4
    $\mathbf{c}_{\mathbf{x}}^* \leftarrow (\mathbf{B}_{\mathbf{x}}\mathbf{B}_{\mathbf{x}}^\top + \lambda\mathbf{I})^{-1}\mathbf{B}_{\mathbf{x}}\mathbf{x}$
    $\hat{\mathbf{x}} \leftarrow \mathbf{B}_{\mathbf{x}}^\top \mathbf{c}_{\mathbf{x}}^*$
    $\rho(\mathbf{x}) \leftarrow \|\mathbf{x} - \hat{\mathbf{x}}\|_2$
end for
$\mathcal{S}_{\text{res}} \leftarrow$ top-min$[m_{\text{res}}, N_{\text{res}}]$ non-anchor tokens by $\rho(\mathbf{x})$ ▷ a shortfall is not returned to m_{anc}
Initialize keep mask from $\mathcal{S}_{\text{anc}} \cup \mathcal{S}_{\text{res}}$
for $t = 0, \ldots, F-1$ **do**
    If fewer than four tokens are kept in frame t, add its first unkept positions up to four
end for
$\tilde{\mathbf{V}} \leftarrow$ kept tokens flattened in original temporal order; $m_{\text{real}} \leftarrow |\tilde{\mathbf{V}}|$
return $\tilde{\mathbf{V}}$

B Full Results and Measurement Scope

B.1 Evaluation Protocol and Uncertainty Contract

Run conditions. The evaluated checkpoints are LLaVA-OneVision-Qwen2 0.5B and 7B (Li et al., 2024a) and Qwen2.5-VL 7B (Bai et al., 2025). Every score path uses the official `lmms-eval` checkout pinned by the repository evaluation lock, model-specific adapters, greedy decoding at batch size 1 and at most 32 sampled frames. The PreViC condition is the fixed full-dimensional post-projector selector. Besides the defaults of Table 7, it uses the adaptive ridge coefficient of Equation 4, and its target rounding and four-token-per-frame floor are in Appendix A.

Benchmark populations. Populations are never pooled across benchmarks. Video-MME resampling retains each video's question triplet, and the other benchmarks use their declared shared video or item unit, so the cluster counts are 3,899 task–video units for MVBench, 900 videos for Video-MME, 753 for LongVideoBench and 500 for EgoSchema.

Budget accounting. The retain ratio is the setting a deployment requests (Section 3). Realized retention coincides with it for PreViC up to its integer rounding, per-frame floor and non-anchor capacity bound, which Table 14 records. Realized counts enter no comparison, ranking or figure axis anywhere in this paper.

Pre-registered evaluation protocol. Whether PreViC preserves the uncompressed model's accuracy is decided by a protocol registered and hashed before any result under it was read, rather than by inspecting point estimates. It fixes a paired cluster bootstrap with $B = 10{,}000$ resamples at seed 42, resampling each benchmark's physical videos, or task–video units for MVBench. It also fixes a non-inferiority margin of $\delta = 1.0$ pp on four-benchmark average accuracy and a two-sided equivalence rule, each decided by where the two-sided 90% interval on the paired difference falls relative to that margin. Non-inferiority is the rule the preservation claim uses. Two-sided equivalence additionally fails on a large *positive* excursion, which is why the two can disagree at the same ratio. The margin follows from the evaluation sizes alone. Writing n_b for the number of scored questions in benchmark b, the contract sets

$$\delta_0 = 100 \cdot \frac{1}{4}\sqrt{\sum_b \frac{0.25}{n_b}}, \tag{3}$$

one worst-case binomial standard error of the four-benchmark average at an accuracy of 0.5, and takes δ to be δ_0 rounded up to the nearest 0.5 pp. The four n_b are 4,000, 2,700, 1,337 and 500 for MVBench, Video-MME, LongVideoBench and EgoSchema, whose separate worst-case standard errors are 0.79, 0.96, 1.37 and 2.24 pp. Together they give $\delta_0 = 0.7255$ pp and $\delta = 1.0$ pp, and the same rule applied to one benchmark at a time gives the per-benchmark margins of 1.0, 1.0, 1.5 and 2.5 pp, which we report but apply to no verdict. These n_b are question counts and are not the cluster counts stated above. No measured quantity enters Equation 3, so the margin is a property of the suite rather than of its outcomes. A binomial standard error also ignores the within-video clustering that the bootstrap resamples, so it understates the true standard error and yields a stricter margin than a cluster-aware derivation would. A smaller margin would assert a resolution these populations do not have. All budgets are compared in Appendix B.5, and the protocol governs only that comparison.

B.2 Full Retain-Ratio Sweeps

This subsection holds the per-coordinate accuracy record behind the main text. Every ratio label is a retain ratio, the budget axis used throughout the paper, and the benchmark populations are defined in Appendix B.1. The tables are described below in the order they appear.

Table 8 summarizes how far PreViC stays ahead of each baseline. Every entry is PreViC minus that baseline in percentage points, averaged over the coordinates the two share, so a positive value means PreViC is ahead and a larger value means it is ahead by more. Each checkpoint column averages the 20 matched retain-ratio and benchmark coordinates a baseline contributes there, and the last column averages all 60. Every entry in the table is positive, so PreViC leads every baseline on every checkpoint. The overall column runs from 1.12 pp against PruneVid, the closest of the ten, to 6.18 pp against SparseVLM, and the per-checkpoint spread is wider still, from 0.43 to 6.96 pp. Taken instead over each baseline's fifteen cells in Table 2, the value is the same, because the four-benchmark average is unweighted.

Table 9 is the per-benchmark form of the main-text comparison, at three of the matched retain ratios, $\alpha \in \{0.2, 0.5, 0.7\}$, and for PreViC and seven of the ten baselines. Its benchmark columns are exactly the corresponding cells of Tables 10–12, and its Avg. column is their unweighted mean, which reproduces the matching cells of Table 2. It is read the same way as that table. The Vanilla row is the uncompressed reference and is never bolded or ranked, and bold marks the highest compressed value in a column. The † markers are the two matched-ratio exceptions that fall at these three budgets, named in Section 4.2. The third exception in the study, PruneVid at $\alpha = 0.1$ on Qwen2.5-VL 7B, lies outside the ratios shown here and is in Table 2. None of the three is separated from PreViC, since all three paired 90% intervals contain zero.

Tables 10–12 give the complete five-ratio grid for all ten baselines alongside PreViC and Vanilla, one table per checkpoint. PreViC's three further budgets, $\alpha \in \{0.4, 0.6, 0.8\}$, are outside this shared grid and appear as four-benchmark averages in Table 16. Dashes denote inapplicable cells, not failed runs. Averaging each row over the four benchmarks gives Table 2, so every other accuracy figure in the paper is derived from these three tables.

Table 13 breaks MVBench into its 20 official tasks at $\alpha = 0.7$, where Δ is PreViC minus Vanilla in percentage points and the final row is the official 20-task mean. Both favorable and unfavorable task differences are shown. In the OV-7B breakdown the action-prediction task rises by 5.5 pp while

Table 8: Mean PreViC–baseline accuracy difference (pp) for all ten baselines, over the 20 matched retain-ratio/benchmark coordinates per checkpoint and 60 overall. Because the four-benchmark average is unweighted, this equals the mean of that baseline's fifteen cells in Table 2.

Baseline	OV-0.5B	OV-7B	Qwen-7B	Overall
DivPrune	0.74	1.49	2.09	1.44
FastVID	0.93	1.70	2.29	1.64
PruneVid	0.43	1.62	1.31	1.12
AIM	0.91	1.49	1.89	1.43
HoliTom	1.48	1.50	2.73	1.90
ApET	0.85	1.61	1.86	1.44
UniComp	0.84	0.85	4.87	2.19
DyCoke	0.60	1.86	3.06	1.84
FastV	1.87	2.97	2.60	2.48
SparseVLM	5.21	6.39	6.96	6.18

Table 9: Per-benchmark accuracy (%) at matched retain ratios, for PreViC and seven of the ten baselines. Avg. is the unweighted mean of the four benchmarks. Bold marks the highest compressed value per column among the methods shown. The Vanilla row carries no bold. †: a matched-ratio exception (Section 4.2).

	MVBench			Video-MME			EgoSchema			LongVideoBench			Avg.		
							retain ratio α								
Method	.20	.50	.70	.20	.50	.70	.20	.50	.70	.20	.50	.70	.20	.50	.70
LLaVA-OV 0.5B															
Vanilla (100%)		46.78			43.81			26.40			47.57			41.14	
DivPrune	44.85	**46.88**	46.88	43.11	44.19	43.74	25.00	27.00	26.00	46.37	46.52	47.42	39.83	**41.15**†	41.01
FastVID	45.75	46.43	**47.03**	42.59	43.59	43.37	25.20	26.00	25.60	45.70	46.75	47.12	39.81	40.69	40.78
PruneVid	45.93	45.80	45.80	**44.67**	44.11	44.22	25.60	25.60	25.40	**47.20**	46.67	47.20	**40.85**†	40.55	40.65
AIM	44.93	45.65	46.25	42.74	43.81	43.78	26.20	26.60	25.40	46.90	46.00	46.67	40.19	40.52	40.52
HoliTom	44.83	46.25	46.45	41.78	42.81	42.89	26.00	25.00	25.20	45.70	46.90	45.03	39.58	40.24	39.89
ApET	45.10	46.38	47.00	43.00	43.11	43.63	27.00	24.40	25.60	45.92	46.82	46.67	40.26	40.18	40.73
UniComp	43.38	44.95	46.13	43.96	44.00	43.70	**27.20**	**28.80**	**28.00**	44.73	45.40	45.85	39.82	40.79	40.92
PreViC	**46.05**	46.78	46.93	43.93	**44.78**	**44.85**	26.60	25.40	26.00	46.75	**47.05**	**48.17**	40.83	41.00	**41.49**
LLaVA-OV 7B															
Vanilla (100%)		58.28			58.41			62.20			56.54			58.86	
DivPrune	56.33	53.85	56.78	56.56	57.48	57.93	61.00	62.80	62.40	55.95	55.95	55.65	57.46	57.52	58.19
FastVID	55.95	55.78	54.55	56.67	57.48	57.30	59.40	60.80	61.00	55.65	57.14	56.39	56.92	57.80	57.31
PruneVid	54.73	54.80	54.73	56.96	58.04	57.78	60.40	61.20	62.00	55.42	56.69	55.57	56.88	57.68	57.52
AIM	55.18	53.58	55.13	56.85	57.89	57.67	62.80	62.20	59.60	57.67	**57.37**	55.20	58.12	57.76	56.90
HoliTom	56.33	55.83	55.10	55.67	56.96	57.04	**65.80**	61.40	61.20	53.55	55.72	56.47	57.84	57.48	57.45
ApET	54.28	52.78	54.85	57.78	57.89	57.37	62.60	61.20	60.00	**58.04**	56.62	56.47	58.17	57.12	57.17
UniComp	54.78	54.58	54.18	**57.85**	58.00	58.22	63.40	**63.60**	**64.20**	56.25	56.17	55.80	58.07	58.09	58.10
PreViC	**58.20**	**58.38**	**58.35**	57.37	**59.26**	**59.30**	62.80	62.40	61.80	56.47	56.62	**56.69**	**58.71**	**59.16**	**59.04**
Qwen2.5-VL 7B															
Vanilla (100%)		68.50			61.41			58.80			58.49			61.80	
DivPrune	63.00	62.88	62.88	55.67	57.19	58.19	**58.80**	58.40	57.00	53.93	54.23	54.75	57.85	58.17	58.20
FastVID	62.50	61.90	61.73	55.41	57.56	57.81	56.00	59.00	58.40	53.18	55.12	54.30	56.77	58.39	58.06
PruneVid	63.45	62.95	62.85	**58.04**	**58.74**	58.96	57.20	57.20	56.80	54.97	54.67	54.82	58.42	58.39	58.36
AIM	62.08	63.60	63.58	56.67	58.19	58.30	56.60	58.20	58.60	54.15	54.38	55.35	57.37	58.59	58.95
HoliTom	61.43	62.20	62.10	54.11	56.07	57.59	55.60	57.80	59.00	52.58	55.05	54.75	55.93	57.78	58.36
ApET	62.40	63.48	63.65	53.52	57.15	58.00	58.60	**60.00**	56.80	55.20	55.35	55.95	57.43	58.99	58.60
UniComp	56.65	57.05	58.18	53.33	53.37	53.33	56.80	57.00	57.00	51.53	51.61	51.76	54.58	54.76	55.07
PreViC	**66.15**	**67.75**	**68.28**	54.07	57.22	**59.48**	58.20	59.60	**59.80**	**57.59**	**58.19**	**58.26**	**59.00**	**60.69**	**61.46**

episodic reasoning falls by 3.5 pp. These 200-question slices are descriptive and are not task-specific significance claims.

B.3 Measured Realized Retention

Table 14 publishes PreViC's measured realized visual-token retention at every retain ratio, pooled over the four benchmarks. The table reports the counts returned by PreViC's own selector. That selector rounds its global target and then applies the four-token-per-frame floor of Appendix A. Ex-

Table 10: LLaVA-OV 0.5B accuracy (%) across retain ratios, all eleven methods with the uncompressed reference. Dashes mark inapplicable cells, not failed runs.

Method	Benchmark	Vanilla	.10	.20	.30	.50	.70
Vanilla	MVBench	46.78	–	–	–	–	–
Vanilla	Video-MME	43.81	–	–	–	–	–
Vanilla	EgoSchema	26.40	–	–	–	–	–
Vanilla	LongVideoBench	47.57	–	–	–	–	–
DivPrune	MVBench	–	44.33	44.85	46.13	46.88	46.88
DivPrune	Video-MME	–	42.26	43.11	43.22	44.19	43.74
DivPrune	EgoSchema	–	26.20	25.00	25.60	27.00	26.00
DivPrune	LongVideoBench	–	44.88	46.37	45.47	46.52	47.42
DyCoke	MVBench	–	45.88	45.88	46.13	45.83	47.18
DyCoke	Video-MME	–	41.96	41.96	42.52	44.44	44.41
DyCoke	EgoSchema	–	26.60	26.60	27.00	25.00	25.20
DyCoke	LongVideoBench	–	45.85	45.85	47.05	47.20	46.37
FastVID	MVBench	–	45.15	45.75	45.98	46.43	47.03
FastVID	Video-MME	–	41.15	42.59	43.04	43.59	43.37
FastVID	EgoSchema	–	25.80	25.20	25.60	26.00	25.60
FastVID	LongVideoBench	–	43.68	45.70	46.75	46.75	47.12
PruneVid	MVBench	–	45.73	45.93	45.88	45.80	45.80
PruneVid	Video-MME	–	43.59	44.67	43.81	44.11	44.22
PruneVid	EgoSchema	–	25.60	25.60	26.20	25.60	25.40
PruneVid	LongVideoBench	–	46.60	47.20	46.75	46.67	47.20
AIM	MVBench	–	43.43	44.93	45.35	45.65	46.25
AIM	Video-MME	–	41.15	42.74	43.30	43.81	43.78
AIM	EgoSchema	–	26.60	26.20	27.20	26.60	25.40
AIM	LongVideoBench	–	44.88	46.90	45.77	46.00	46.67
HoliTom	MVBench	–	43.78	44.83	45.33	46.25	46.45
HoliTom	Video-MME	–	39.44	41.78	41.74	42.81	42.89
HoliTom	EgoSchema	–	27.20	26.00	25.80	25.00	25.20
HoliTom	LongVideoBench	–	43.31	45.70	45.85	46.90	45.03
FastV	MVBench	–	43.95	45.38	45.95	46.78	46.68
FastV	Video-MME	–	37.96	39.70	41.44	43.19	43.30
FastV	EgoSchema	–	23.40	24.40	25.20	24.80	25.40
FastV	LongVideoBench	–	43.68	45.77	45.70	45.03	45.77
SparseVLM	MVBench	–	42.15	41.83	41.85	42.08	42.00
SparseVLM	Video-MME	–	37.19	37.37	37.67	37.63	37.26
SparseVLM	EgoSchema	–	22.40	22.40	23.20	22.80	22.40
SparseVLM	LongVideoBench	–	41.06	41.59	41.29	41.14	41.44
ApET	MVBench	–	44.78	45.10	45.85	46.38	47.00
ApET	Video-MME	–	42.00	43.00	42.89	43.11	43.63
ApET	EgoSchema	–	26.00	27.00	25.60	24.40	25.60
ApET	LongVideoBench	–	45.33	45.92	46.82	46.82	46.67
UniComp	MVBench	–	42.90	43.38	43.88	44.95	46.13
UniComp	Video-MME	–	42.78	43.96	43.93	44.00	43.70
UniComp	EgoSchema	–	26.20	27.20	28.20	28.80	28.00
UniComp	LongVideoBench	–	45.47	44.73	44.65	45.40	45.85
PreViC	MVBench	–	45.50	46.05	46.55	46.78	46.93
PreViC	Video-MME	–	44.07	43.93	44.04	44.78	44.85
PreViC	EgoSchema	–	26.80	26.60	27.20	25.40	26.00
PreViC	LongVideoBench	–	46.82	46.75	46.60	47.05	48.17

actly two mechanisms make the returned count m_{real} depart from that target. The floor raises it, and the non-anchor capacity bound N_{res} truncates a shortfall instead of returning it to the anchor budget, which lowers it on short inputs.

B.4 Hyperparameter Sensitivity

Table 15 varies one PreViC setting at a time around the defaults of Table 7, on MVBench with LLaVA-OV 7B at $\alpha = 0.3$. All rows come from one run family on the full 4,000-question population, so they are directly comparable to one another.

The method is insensitive to the two settings of its predictor. Moving the neighborhood size k_{nn} across an eightfold range costs at most 0.28 pp, and replacing the adaptive ridge coefficient of Equation 4 with either fixed scalar costs 0.13 pp. Neither is a tuned quantity.

The budget split β is the setting with the largest effect, and it acts in only one direction. Removing the anchor share entirely costs 1.28 pp, the largest single-axis cost we measure, while raising it from 0.3 to 0.5 changes nothing at all. The asymmetry follows from the anchor-pool cap N_{anc} of Section 3. At

Table 11: LLaVA-OV 7B accuracy (%) across retain ratios, all eleven methods with the uncompressed reference. Dashes mark inapplicable cells, not failed runs.

Method	Benchmark	Vanilla	.10	.20	.30	.50	.70
Vanilla	MVBench	58.28	–	–	–	–	–
Vanilla	Video-MME	58.41	–	–	–	–	–
Vanilla	EgoSchema	62.20	–	–	–	–	–
Vanilla	LongVideoBench	56.54	–	–	–	–	–
DivPrune	MVBench	–	52.65	56.33	56.40	53.85	56.78
DivPrune	Video-MME	–	54.41	56.56	56.56	57.48	57.93
DivPrune	EgoSchema	–	61.40	61.00	60.20	62.80	62.40
DivPrune	LongVideoBench	–	53.10	55.95	57.52	55.95	55.65
DyCoke	MVBench	–	57.50	57.50	55.88	53.20	56.25
DyCoke	Video-MME	–	54.33	54.33	54.93	57.89	58.00
DyCoke	EgoSchema	–	60.00	60.00	59.00	61.60	61.80
DyCoke	LongVideoBench	–	53.85	53.85	55.42	56.32	55.72
FastVID	MVBench	–	55.80	55.95	54.95	55.78	54.55
FastVID	Video-MME	–	54.67	56.67	57.22	57.48	57.30
FastVID	EgoSchema	–	58.80	59.40	60.40	60.80	61.00
FastVID	LongVideoBench	–	55.12	55.65	55.65	57.14	56.39
PruneVid	MVBench	–	53.83	54.73	54.70	54.80	54.73
PruneVid	Video-MME	–	56.81	56.96	57.52	58.04	57.78
PruneVid	EgoSchema	–	59.40	60.40	61.00	61.20	62.00
PruneVid	LongVideoBench	–	55.05	55.42	55.65	56.69	55.57
AIM	MVBench	–	50.65	55.18	57.10	53.58	55.13
AIM	Video-MME	–	54.52	56.85	57.11	57.89	57.67
AIM	EgoSchema	–	61.20	62.80	61.20	62.20	59.60
AIM	LongVideoBench	–	54.53	57.67	57.37	57.37	55.20
HoliTom	MVBench	–	53.75	56.33	56.65	55.83	55.10
HoliTom	Video-MME	–	52.78	55.67	58.07	56.96	57.04
HoliTom	EgoSchema	–	61.60	65.80	62.60	61.40	61.20
HoliTom	LongVideoBench	–	52.43	53.55	55.72	55.72	56.47
FastV	MVBench	–	49.95	51.98	53.13	53.90	53.85
FastV	Video-MME	–	51.74	54.89	55.93	57.26	57.30
FastV	EgoSchema	–	58.00	59.60	60.20	60.80	60.80
FastV	LongVideoBench	–	51.23	54.60	56.54	56.92	56.69
SparseVLM	MVBench	–	49.43	49.40	49.38	49.70	49.25
SparseVLM	Video-MME	–	51.22	51.22	50.96	51.07	51.11
SparseVLM	EgoSchema	–	56.80	56.60	57.00	56.80	56.20
SparseVLM	LongVideoBench	–	51.83	52.06	51.98	52.21	52.58
ApET	MVBench	–	50.63	54.28	56.10	52.78	54.85
ApET	Video-MME	–	54.89	57.78	57.37	57.89	57.37
ApET	EgoSchema	–	61.20	62.60	60.40	61.20	60.00
ApET	LongVideoBench	–	55.20	58.04	56.77	56.62	56.47
UniComp	MVBench	–	52.93	54.78	55.83	54.58	54.18
UniComp	Video-MME	–	56.26	57.85	58.07	58.00	58.22
UniComp	EgoSchema	–	62.80	63.40	63.80	63.60	64.20
UniComp	LongVideoBench	–	54.60	56.25	56.25	56.17	55.80
PreViC	MVBench	–	57.80	58.20	58.15	58.38	58.35
PreViC	Video-MME	–	54.78	57.37	58.22	59.26	59.30
PreViC	EgoSchema	–	61.60	62.80	63.20	62.40	61.80
PreViC	LongVideoBench	–	55.27	56.47	57.97	56.62	56.69

this budget the anchor share is already large enough to saturate the anchor frames, so further increases have nowhere to go, whereas removing it strands the hierarchical split that Section 3 relies on.

Replacing the predictive residual with the distance to the single nearest anchor costs 0.70 pp, consistent with Section 4.6, where removing the residual ranking is the more expensive of the two changes.

B.5 Preservation Test: All Budgets

Table 16 prints the 24 PreViC-versus-uncompressed budgets of the frozen contract, and Section 4.3 states the verdicts it establishes and the two rules that must not be conflated. Two properties of the table are restated here. Its NI and EQ columns are interval-based verdicts, strictly stronger than asking whether the point estimate Δ alone falls within the 1.0 pp margin, which is why Qwen2.5-VL at $\alpha = 0.6$ has $\Delta = -0.739$ pp inside a 1.0 pp band and still fails. A positive Δ in this table is evidence that accuracy is preserved, never a gain over the uncompressed model. The two-sided EQ column fails at LLaVA-OV 7B $\alpha = 0.3$ and 0.4 precisely because the differences there are positive and their intervals reach past +1.0 pp.

Table 12: Qwen2.5-VL 7B accuracy (%) across retain ratios, all eleven methods with the uncompressed reference. Dashes mark inapplicable cells, not failed runs.

Method	Benchmark	Vanilla	.10	.20	.30	.50	.70
Vanilla	MVBench	68.50	–	–	–	–	–
Vanilla	Video-MME	61.41	–	–	–	–	–
Vanilla	EgoSchema	58.80	–	–	–	–	–
Vanilla	LongVideoBench	58.49	–	–	–	–	–
DivPrune	MVBench	–	60.93	63.00	63.28	62.88	62.88
DivPrune	Video-MME	–	53.15	55.67	56.15	57.19	58.19
DivPrune	EgoSchema	–	55.20	58.80	57.60	58.40	57.00
DivPrune	LongVideoBench	–	52.06	53.93	54.53	54.23	54.75
DyCoke	MVBench	–	61.78	61.78	61.80	63.23	63.35
DyCoke	Video-MME	–	52.22	52.22	52.22	57.44	58.44
DyCoke	EgoSchema	–	56.20	56.20	56.20	58.60	58.60
DyCoke	LongVideoBench	–	50.41	50.41	50.41	53.85	54.97
FastVID	MVBench	–	61.28	62.50	62.33	61.90	61.73
FastVID	Video-MME	–	52.81	55.41	56.37	57.56	57.81
FastVID	EgoSchema	–	55.60	56.00	59.40	59.00	58.40
FastVID	LongVideoBench	–	51.38	53.18	53.63	55.12	54.30
PruneVid	MVBench	–	63.23	63.45	63.13	62.95	62.85
PruneVid	Video-MME	–	57.74	58.04	58.41	58.74	58.96
PruneVid	EgoSchema	–	56.80	57.20	57.00	57.20	56.80
PruneVid	LongVideoBench	–	54.15	54.97	54.30	54.67	54.82
AIM	MVBench	–	60.25	62.08	63.03	63.60	63.58
AIM	Video-MME	–	54.22	56.67	57.22	58.19	58.30
AIM	EgoSchema	–	55.80	56.60	57.20	58.20	58.60
AIM	LongVideoBench	–	51.98	54.15	54.53	54.38	55.35
HoliTom	MVBench	–	60.78	61.43	62.35	62.20	62.10
HoliTom	Video-MME	–	52.19	54.11	54.96	56.07	57.59
HoliTom	EgoSchema	–	55.00	55.60	58.60	57.80	59.00
HoliTom	LongVideoBench	–	51.01	52.58	53.78	55.05	54.75
FastV	MVBench	–	58.20	60.90	62.18	63.25	63.38
FastV	Video-MME	–	53.74	56.22	56.93	58.22	59.07
FastV	EgoSchema	–	55.80	57.40	59.40	57.60	58.20
FastV	LongVideoBench	–	47.49	51.01	51.68	54.30	54.60
SparseVLM	MVBench	–	57.18	56.78	57.20	57.13	57.20
SparseVLM	Video-MME	–	51.37	51.00	51.22	51.07	50.93
SparseVLM	EgoSchema	–	52.60	52.80	53.20	53.20	53.40
SparseVLM	LongVideoBench	–	49.07	49.29	49.59	49.36	48.92
ApET	MVBench	–	61.20	62.40	63.20	63.48	63.65
ApET	Video-MME	–	52.26	53.52	54.89	57.15	58.00
ApET	EgoSchema	–	56.80	58.60	58.20	60.00	56.80
ApET	LongVideoBench	–	52.58	55.20	55.27	55.35	55.95
UniComp	MVBench	–	56.48	56.65	56.70	57.05	58.18
UniComp	Video-MME	–	53.19	53.33	53.33	53.37	53.33
UniComp	EgoSchema	–	57.00	56.80	56.80	57.00	57.00
UniComp	LongVideoBench	–	51.53	51.53	51.61	51.61	51.76
PreViC	MVBench	–	64.40	66.15	66.90	67.75	68.28
PreViC	Video-MME	–	52.30	54.07	55.52	57.22	59.48
PreViC	EgoSchema	–	55.80	58.20	58.40	59.60	59.80
PreViC	LongVideoBench	–	56.02	57.59	57.67	58.19	58.26

Tables 17–19 apply the same contract and estimator to every baseline, all 150 coordinates of 10 methods, 3 checkpoints and the 5 shared retain ratios. They are the evidence behind the counts of Table 3 and behind the statement in Section 4.3 that no baseline matches on either 7B checkpoint. Eight coordinates are non-inferior and the same eight are equivalent, all of them on LLaVA-OV 0.5B. On LLaVA-OV 7B the nearest is UniComp at $\alpha = 0.3$, whose interval still reaches -1.237 pp, and on Qwen2.5-VL 7B the nearest is ApET at $\alpha = 0.5$, whose whole interval lies below the margin.

The same estimator applied to the PreViC-versus-baseline differences covers those same 150 coordinates. Its interval lies entirely above zero at 127 of them and contains zero at the remaining 23, and at none of the 150 does it lie entirely below zero, so no baseline is separated from PreViC in its own favor anywhere. This includes the three settings where a baseline's matched-ratio average is higher, whose intervals are $[-0.721, +0.678]$ for PruneVid at $\alpha = 0.2$ on LLaVA-OV 0.5B, $[-0.653, +0.344]$ for DivPrune at 0.5 on the same checkpoint and $[-1.963, +0.251]$ for PruneVid at 0.1 on Qwen2.5-VL 7B. The last of these is the most baseline-favorable coordinate in the grid and its interval still reaches +0.251 pp. Against the strongest baseline at each of the twelve settings where PreViC has the highest average, the interval lies above zero at $\alpha = 0.3$, 0.5 and 0.7 on both 7B checkpoints and contains zero at the other six, which are the three LLaVA-OV 0.5B wins, LLaVA-OV 7B at $\alpha = 0.1$ and 0.2 and Qwen2.5-VL 7B at 0.2.

Table 13: MVBench per-task accuracy (%) at retain ratio 0.7. Δ is PreViC minus Vanilla (pp). The final row is the official 20-task mean.

Task	OV-0.5B			OV-7B			Qwen-7B		
	Vanilla	PreViC	Δ	Vanilla	PreViC	Δ	Vanilla	PreViC	Δ
action antonym	55.00	55.50	+0.50	80.50	77.50	−3.00	86.00	86.50	+0.50
action count	39.50	39.00	−0.50	48.00	49.00	+1.00	47.00	47.00	+0.00
action localization	35.00	35.50	+0.50	56.50	55.50	−1.00	59.50	57.00	−2.50
action prediction	43.00	45.00	+2.00	56.00	61.50	+5.50	68.50	70.00	+1.50
action sequence	56.50	57.00	+0.50	71.00	73.00	+2.00	81.00	82.00	+1.00
character order	55.00	53.00	−2.00	71.00	73.50	+2.50	80.00	79.50	−0.50
counterfactual inference	37.50	41.00	+3.50	50.50	52.00	+1.50	71.00	71.50	+0.50
egocentric navigation	31.50	32.00	+0.50	33.00	34.00	+1.00	41.50	39.00	−2.50
episodic reasoning	44.50	44.50	+0.00	57.00	53.50	−3.50	51.50	52.50	+1.00
fine-grained action	39.50	38.50	−1.00	47.50	46.50	−1.00	49.00	47.50	−1.50
fine-grained pose	50.00	49.00	−1.00	55.00	53.50	−1.50	70.00	70.50	+0.50
moving attribute	53.00	51.00	−2.00	70.50	71.00	+0.50	92.50	93.00	+0.50
moving count	34.00	30.00	−4.00	48.00	45.50	−2.50	80.50	78.00	−2.50
moving direction	21.00	22.00	+1.00	22.00	23.00	+1.00	49.00	49.00	+0.00
object existence	47.50	51.00	+3.50	58.00	56.50	−1.50	92.50	92.50	+0.00
object interaction	69.50	71.50	+2.00	82.50	83.00	+0.50	81.50	81.50	+0.00
object shuffle	32.00	32.50	+0.50	35.50	34.00	−1.50	36.00	37.50	+1.50
scene transition	84.00	82.50	−1.50	93.00	93.00	+0.00	93.00	92.00	−1.00
state change	37.50	39.50	+2.00	51.50	52.00	+0.50	60.00	61.50	+1.50
unexpected action	70.00	68.50	−1.50	78.50	79.50	+1.00	80.00	77.50	−2.50
Avg.	46.775	46.925	+0.150	58.275	58.350	+0.075	68.500	68.275	−0.225

Table 14: PreViC's measured realized visual-token retention (%), pooled over the four benchmarks, against the requested retain ratio α.

	retain ratio α							
Checkpoint	.10	.20	.30	.40	.50	.60	.70	.80
LLaVA-OV 0.5B	10.11	20.04	30.03	40.02	50.01	60.00	70.00	80.01
LLaVA-OV 7B	10.08	20.03	30.02	40.01	50.01	60.00	70.00	80.01
Qwen2.5-VL 7B	10.00	20.00	30.00	40.00	50.00	60.00	70.00	80.00

Table 15: PreViC hyperparameter sensitivity on MVBench with LLaVA-OV 7B at $\alpha = 0.3$. Each row changes one setting away from the default configuration and leaves the others at their defaults. Δ is the MVBench accuracy of that row minus the accuracy of the default configuration measured in the same run family, so a negative value means the change costs accuracy.

Setting	Value	Δ (pp)
Default	$a = 2$, $\beta = 0.3$, $k_{\text{nn}} = 4$, adaptive λ, residual	0.00
Anchor count a	1	−0.15
	3	−0.50
Anchor budget share β	0.0	−1.28
	0.15	−0.53
	0.5	0.00
Neighbors k_{nn}	1	−0.08
	2	−0.15
	8	−0.28
Ridge λ	0.001	−0.13
	0.01	−0.13
Importance signal	nearest anchor	−0.70

Table 20 decomposes the $\alpha = 0.5$ PreViC–Vanilla contrast of Table 16 by benchmark, under the same protocol and estimator. The Video-MME clusters are physical videos, each carrying its three questions. These are descriptive per-benchmark intervals, and no verdict is read off a single benchmark. Nine of the twelve contain zero. The three that do not are the two Qwen2.5-VL 7B losses on Video-MME and MVBench, the first of which, at 4.19 pp, drives that checkpoint's shortfall at this ratio, and a LLaVA-OV 0.5B Video-MME interval whose lower end is +0.04 pp.

Table 16: All 24 PreViC-versus-uncompressed budgets under the pre-registered equivalence protocol. Δ is average accuracy PreViC minus uncompressed (pp), with its paired bootstrap CI90. NI marks one-sided non-inferiority at the 1.0 pp margin, EQ the stricter two-sided equivalence.

Checkpoint	α	PreViC	Vanilla	Δ (pp)	CI90 (pp)	NI / EQ
LLaVA-OV 0.5B	.10	40.80	41.14	−0.341	[−1.165, +0.462]	– / –
	.20	40.83	41.14	−0.309	[−1.040, +0.423]	– / –
	.30	41.10	41.14	−0.044	[−0.794, +0.687]	✓/ ✓
	.40	40.69	41.14	−0.454	[−1.090, +0.189]	– / –
	.50	41.00	41.14	−0.140	[−0.765, +0.465]	✓/ ✓
	.60	40.96	41.14	−0.179	[−0.750, +0.398]	✓/ ✓
	.70	41.49	41.14	+0.346	[−0.236, +0.909]	✓/ ✓
	.80	40.93	41.14	−0.211	[−0.723, +0.303]	✓/ ✓
LLaVA-OV 7B	.10	57.36	58.86	−1.494	[−2.390, −0.608]	– / –
	.20	58.71	58.86	−0.147	[−0.926, +0.637]	✓/ ✓
	.30	59.38	58.86	+0.528	[−0.212, +1.293]	✓/ –
	.40	59.33	58.86	+0.474	[−0.204, +1.183]	✓/ –
	.50	59.16	58.86	+0.307	[−0.294, +0.906]	✓/ ✓
	.60	58.65	58.86	−0.211	[−0.796, +0.373]	✓/ ✓
	.70	59.04	58.86	+0.178	[−0.339, +0.701]	✓/ ✓
	.80	59.14	58.86	+0.279	[−0.217, +0.785]	✓/ ✓
Qwen2.5-VL 7B	.10	57.13	61.80	−4.670	[−5.573, −3.759]	– / –
	.20	59.00	61.80	−2.795	[−3.628, −1.952]	– / –
	.30	59.62	61.80	−2.178	[−2.941, −1.408]	– / –
	.40	60.15	61.80	−1.647	[−2.326, −0.957]	– / –
	.50	60.69	61.80	−1.109	[−1.750, −0.481]	– / –
	.60	61.06	61.80	−0.739	[−1.358, −0.116]	– / –
	.70	61.46	61.80	−0.344	[−0.863, +0.185]	✓/ ✓
	.80	61.71	61.80	−0.090	[−0.514, +0.350]	✓/ ✓

B.6 Exploratory Multi-Query Timing

Table 21 gives the exact values plotted in Figure 4, measured on LLaVA-OV 7B at $\alpha = 0.5$ on an NVIDIA RTX 6000 Ada Generation GPU. Each row is a query count n for one video, and the two time columns are cumulative first-token LLM time under the two ways of serving those n questions. Recompute rebuilds the visual prefix for every question, while Visual KV builds it once and reuses it, which is available to PreViC because the compressed prefix does not depend on the question. Reuse is what a multi-query workload buys, so the table is read from $n \geq 2$, where the visual-KV time includes the one-time 0.461-s prefix build.

The two paths separate immediately and the absolute gap keeps widening, reaching 462.9 s at 1,024 queries, where recomputation costs 497.087 s against 34.158 s under reuse. The ratio grows with n without approaching the number of queries, passing 2.778× at $n = 4$ and an order of magnitude by $n = 64$. The ratio stays below the number of queries because only the visual prefix is shared, and each question still carries its own tokens through the language model. This is a diagnostic rather than a latency result. It starts from prepared embeddings and omits video preparation, vision, projection, selection and full-request cost, so the ratios above do not transfer to end-to-end serving, where Table 5 shows the LLM prefill is a minority of the measured time. Both plotted series are PreViC, and no other method was run in this diagnostic.

Table 17: Baseline-versus-uncompressed budgets on LLaVA-OV 0.5B, under the same protocol as Table 16. Accuracy is the four-benchmark average (%) and Δ is baseline minus uncompressed average accuracy (pp) with its paired bootstrap CI90. The uncompressed reference is the Vanilla row. NI marks one-sided non-inferiority at the 1.0 pp margin, EQ the stricter two-sided equivalence.

Method	α	Accuracy	Δ (pp)	CI90 (pp)	NI / EQ
Vanilla	–	41.14	–	–	–
FastV	.10	37.25	−3.892	[−4.693, −3.130]	– / –
	.20	38.81	−2.327	[−3.042, −1.603]	– / –
	.30	39.57	−1.566	[−2.258, −0.914]	– / –
	.50	39.95	−1.193	[−1.793, −0.608]	– / –
	.70	40.29	−0.853	[−1.333, −0.369]	– / –
DivPrune	.10	39.42	−1.725	[−2.503, −0.957]	– / –
	.20	39.83	−1.306	[−2.055, −0.562]	– / –
	.30	40.11	−1.034	[−1.766, −0.311]	– / –
	.50	41.15	+0.006	[−0.613, +0.625]	✓/ ✓
	.70	41.01	−0.131	[−0.722, +0.445]	✓/ ✓
DyCoke	.10	40.07	−1.068	[−1.791, −0.348]	– / –
	.20	40.07	−1.068	[−1.795, −0.342]	– / –
	.30	40.67	−0.467	[−1.179, +0.257]	– / –
	.50	40.62	−0.524	[−1.120, +0.073]	– / –
	.70	40.79	−0.351	[−0.902, +0.197]	✓/ ✓
AIM	.10	39.01	−2.127	[−2.940, −1.319]	– / –
	.20	40.19	−0.949	[−1.673, −0.240]	– / –
	.30	40.41	−0.735	[−1.457, −0.028]	– / –
	.50	40.52	−0.624	[−1.197, −0.055]	– / –
	.70	40.52	−0.615	[−1.157, −0.079]	– / –
SparseVLM	.10	35.70	−5.440	[−6.309, −4.593]	– / –
	.20	35.80	−5.344	[−6.194, −4.493]	– / –
	.30	36.00	−5.139	[−5.984, −4.299]	– / –
	.50	35.91	−5.229	[−6.045, −4.404]	– / –
	.70	35.77	−5.366	[−6.236, −4.512]	– / –
PruneVid	.10	40.38	−0.761	[−1.432, −0.106]	– / –
	.20	40.85	−0.293	[−0.920, +0.318]	✓/ ✓
	.30	40.66	−0.481	[−1.083, +0.126]	– / –
	.50	40.55	−0.594	[−1.176, −0.006]	– / –
	.70	40.65	−0.485	[−1.083, +0.106]	– / –
FastVID	.10	38.94	−2.195	[−3.067, −1.342]	– / –
	.20	39.81	−1.329	[−2.069, −0.579]	– / –
	.30	40.34	−0.800	[−1.456, −0.165]	– / –
	.50	40.69	−0.449	[−0.982, +0.089]	✓/ ✓
	.70	40.78	−0.361	[−0.802, +0.074]	✓/ ✓
HoliTom	.10	38.43	−2.708	[−3.560, −1.850]	– / –
	.20	39.58	−1.564	[−2.318, −0.823]	– / –
	.30	39.68	−1.461	[−2.221, −0.718]	– / –
	.50	40.24	−0.900	[−1.570, −0.245]	– / –
	.70	39.89	−1.248	[−1.859, −0.631]	– / –
ApET	.10	39.53	−1.615	[−2.416, −0.826]	– / –
	.20	40.26	−0.884	[−1.585, −0.172]	– / –
	.30	40.29	−0.850	[−1.518, −0.197]	– / –
	.50	40.18	−0.963	[−1.572, −0.360]	– / –
	.70	40.73	−0.414	[−0.970, +0.135]	✓/ ✓
UniComp	.10	39.34	−1.802	[−2.670, −0.944]	– / –
	.20	39.82	−1.324	[−2.122, −0.537]	– / –
	.30	40.16	−0.976	[−1.808, −0.114]	– / –
	.50	40.79	−0.352	[−1.167, +0.430]	– / –
	.70	40.92	−0.220	[−0.995, +0.554]	✓/ ✓

Table 18: Baseline-versus-uncompressed budgets on LLaVA-OV 7B, under the same protocol as Table 16. Accuracy is the four-benchmark average (%) and Δ is baseline minus uncompressed average accuracy (pp) with its paired bootstrap CI90. The uncompressed reference is the Vanilla row. NI marks one-sided non-inferiority at the 1.0 pp margin, EQ the stricter two-sided equivalence.

Method	α	Accuracy	Δ (pp)	CI90 (pp)	NI / EQ
Vanilla	–	58.86	–	–	–
FastV	.10	52.73	−6.126	[−7.124, −5.131]	– / –
	.20	55.27	−3.591	[−4.467, −2.731]	– / –
	.30	56.45	−2.408	[−3.209, −1.624]	– / –
	.50	57.22	−1.637	[−2.323, −0.974]	– / –
	.70	57.16	−1.697	[−2.331, −1.079]	– / –
DivPrune	.10	55.39	−3.466	[−4.444, −2.510]	– / –
	.20	57.46	−1.400	[−2.278, −0.536]	– / –
	.30	57.67	−1.189	[−1.988, −0.359]	– / –
	.50	57.52	−1.337	[−2.098, −0.592]	– / –
	.70	58.19	−0.670	[−1.315, −0.040]	– / –
DyCoke	.10	56.42	−2.435	[−3.327, −1.528]	– / –
	.20	56.42	−2.435	[−3.318, −1.575]	– / –
	.30	56.31	−2.551	[−3.400, −1.681]	– / –
	.50	57.25	−1.604	[−2.392, −0.833]	– / –
	.70	57.94	−0.914	[−1.605, −0.245]	– / –
AIM	.10	55.22	−3.633	[−4.567, −2.682]	– / –
	.20	58.12	−0.733	[−1.567, +0.110]	– / –
	.30	58.19	−0.662	[−1.456, +0.133]	– / –
	.50	57.76	−1.099	[−1.864, −0.362]	– / –
	.70	56.90	−1.959	[−2.636, −1.299]	– / –
SparseVLM	.10	52.32	−6.537	[−7.612, −5.485]	– / –
	.20	52.32	−6.537	[−7.594, −5.493]	– / –
	.30	52.33	−6.527	[−7.593, −5.451]	– / –
	.50	52.45	−6.412	[−7.458, −5.357]	– / –
	.70	52.29	−6.571	[−7.648, −5.535]	– / –
PruneVid	.10	56.27	−2.585	[−3.438, −1.746]	– / –
	.20	56.88	−1.979	[−2.727, −1.242]	– / –
	.30	57.22	−1.640	[−2.340, −0.925]	– / –
	.50	57.68	−1.174	[−1.849, −0.499]	– / –
	.70	57.52	−1.338	[−2.004, −0.671]	– / –
FastVID	.10	56.10	−2.759	[−3.630, −1.873]	– / –
	.20	56.92	−1.941	[−2.756, −1.116]	– / –
	.30	57.05	−1.802	[−2.549, −1.036]	– / –
	.50	57.80	−1.057	[−1.713, −0.409]	– / –
	.70	57.31	−1.546	[−2.199, −0.905]	– / –
HoliTom	.10	55.14	−3.717	[−4.685, −2.742]	– / –
	.20	57.84	−1.021	[−1.902, −0.129]	– / –
	.30	58.26	−0.595	[−1.441, +0.272]	– / –
	.50	57.48	−1.379	[−2.157, −0.593]	– / –
	.70	57.45	−1.405	[−2.165, −0.673]	– / –
ApET	.10	55.48	−3.379	[−4.273, −2.472]	– / –
	.20	58.17	−0.683	[−1.561, +0.199]	– / –
	.30	57.66	−1.197	[−2.026, −0.342]	– / –
	.50	57.12	−1.736	[−2.529, −0.974]	– / –
	.70	57.17	−1.684	[−2.370, −1.015]	– / –
UniComp	.10	56.65	−2.211	[−3.241, −1.213]	– / –
	.20	58.07	−0.789	[−1.678, +0.127]	– / –
	.30	58.49	−0.371	[−1.237, +0.495]	– / –
	.50	58.09	−0.770	[−1.615, +0.079]	– / –
	.70	58.10	−0.758	[−1.600, +0.083]	– / –

Table 19: Baseline-versus-uncompressed budgets on Qwen2.5-VL 7B, under the same protocol as Table 16. Accuracy is the four-benchmark average (%) and Δ is baseline minus uncompressed average accuracy (pp) with its paired bootstrap CI90. The uncompressed reference is the Vanilla row. NI marks one-sided non-inferiority at the 1.0 pp margin, EQ the stricter two-sided equivalence.

Method	α	Accuracy	Δ (pp)	CI90 (pp)	NI / EQ
Vanilla	–	61.80	–	–	–
FastV	.10	53.81	−7.990	[−9.086, −6.868]	– / –
	.20	56.38	−5.416	[−6.465, −4.391]	– / –
	.30	57.55	−4.253	[−5.260, −3.264]	– / –
	.50	58.34	−3.456	[−4.418, −2.482]	– / –
	.70	58.81	−2.987	[−3.895, −2.058]	– / –
DivPrune	.10	55.33	−6.467	[−7.521, −5.409]	– / –
	.20	57.85	−3.951	[−4.961, −2.958]	– / –
	.30	57.89	−3.912	[−4.918, −2.920]	– / –
	.50	58.17	−3.628	[−4.569, −2.681]	– / –
	.70	58.20	−3.597	[−4.526, −2.655]	– / –
DyCoke	.10	55.15	−6.647	[−7.682, −5.601]	– / –
	.20	55.15	−6.647	[−7.693, −5.597]	– / –
	.30	55.16	−6.641	[−7.688, −5.604]	– / –
	.50	58.28	−3.519	[−4.451, −2.551]	– / –
	.70	58.84	−2.957	[−3.920, −1.997]	– / –
AIM	.10	55.56	−6.236	[−7.302, −5.157]	– / –
	.20	57.37	−4.426	[−5.510, −3.353]	– / –
	.30	57.99	−3.806	[−4.779, −2.828]	– / –
	.50	58.59	−3.209	[−4.189, −2.255]	– / –
	.70	58.95	−2.844	[−3.791, −1.901]	– / –
SparseVLM	.10	52.55	−9.247	[−10.375, −8.140]	– / –
	.20	52.47	−9.333	[−10.456, −8.183]	– / –
	.30	52.80	−8.996	[−10.144, −7.848]	– / –
	.50	52.69	−9.108	[−10.224, −7.982]	– / –
	.70	52.61	−9.189	[−10.338, −8.062]	– / –
PruneVid	.10	57.98	−3.820	[−4.858, −2.772]	– / –
	.20	58.42	−3.384	[−4.382, −2.399]	– / –
	.30	58.21	−3.591	[−4.552, −2.631]	– / –
	.50	58.39	−3.408	[−4.348, −2.449]	– / –
	.70	58.36	−3.440	[−4.406, −2.484]	– / –
FastVID	.10	55.27	−6.531	[−7.605, −5.452]	– / –
	.20	56.77	−5.028	[−6.064, −3.987]	– / –
	.30	57.93	−3.868	[−4.897, −2.816]	– / –
	.50	58.39	−3.404	[−4.396, −2.388]	– / –
	.70	58.06	−3.739	[−4.729, −2.729]	– / –
HoliTom	.10	54.74	−7.057	[−8.166, −5.946]	– / –
	.20	55.93	−5.870	[−6.933, −4.793]	– / –
	.30	57.42	−4.377	[−5.422, −3.364]	– / –
	.50	57.78	−4.018	[−5.002, −3.011]	– / –
	.70	58.36	−3.439	[−4.415, −2.456]	– / –
ApET	.10	55.71	−6.089	[−7.117, −5.069]	– / –
	.20	57.43	−4.370	[−5.408, −3.328]	– / –
	.30	57.89	−3.909	[−4.915, −2.917]	– / –
	.50	58.99	−2.806	[−3.778, −1.817]	– / –
	.70	58.60	−3.200	[−4.151, −2.250]	– / –
UniComp	.10	54.55	−7.251	[−8.324, −6.184]	– / –
	.20	54.58	−7.220	[−8.323, −6.109]	– / –
	.30	54.61	−7.189	[−8.303, −6.100]	– / –
	.50	54.76	−7.042	[−8.106, −5.961]	– / –
	.70	55.07	−6.733	[−7.816, −5.670]	– / –

Table 20: Paired PreViC–Vanilla differences at retain ratio 0.5, per benchmark. PreViC accuracy is in %, and Δ and its interval are in pp. The paired 90% intervals come from the protocol of Table 16.

Checkpoint	Benchmark	PreViC	Δ	Paired 90% CI
OV-0.5B	MVBench	46.78	+0.00	[−0.80, +0.83]
	Video-MME	44.78	+0.96	[+0.04, +1.89]
	EgoSchema	25.40	−1.00	[−2.60, +0.60]
	LongVideoBench	47.05	−0.52	[−1.83, +0.77]
OV-7B	MVBench	58.38	+0.10	[−0.62, +0.82]
	Video-MME	59.26	+0.85	[−0.07, +1.78]
	EgoSchema	62.40	+0.20	[−1.40, +1.80]
	LongVideoBench	56.62	+0.07	[−1.33, +1.43]
Qwen-7B	MVBench	67.75	−0.75	[−1.38, −0.12]
	Video-MME	57.22	−4.19	[−5.30, −3.07]
	EgoSchema	59.60	+0.80	[−1.00, +2.60]
	LongVideoBench	58.19	−0.30	[−1.45, +0.87]

Table 21: Cumulative first-token LLM time (s) for Figure 4, LLaVA-OV 7B at $\alpha = 0.5$ on an NVIDIA RTX 6000 Ada Generation GPU. Recompute rebuilds the visual prefix for every question and Visual KV builds it once.

Queries	Recompute	Visual KV	Ratio
1	0.399	0.399	1.000×
2	0.817	0.535	1.526×
4	1.673	0.602	2.778×
8	3.394	0.735	4.619×
16	6.848	0.998	6.859×
32	13.861	1.525	9.086×
64	28.236	2.580	10.943×
128	58.168	4.689	12.406×
256	121.371	8.906	13.628×
512	251.774	17.338	14.522×
1,024	497.087	34.158	14.553×

C Qualitative Visualization of R1 and R2

Figure 5 shows the two stages on one 16-frame video at $\alpha = 0.3$, with the original frames, the R2 predictive-residual map and the kept patches of every frame and the two R1 anchor frames labeled in red. R1 selects the first and fifteenth frames as anchors using frame-level novelty. R2 then concentrates more of the remaining budget on moving objects and human–object interactions than on the static background that the anchor memory already predicts. This remains an illustration of the intended mechanism read off one example, not a benchmark-level causal attribution. A small residual indicates that local anchor tokens reconstruct a projected feature, not that a token is semantically irrelevant to a question (Section 5).

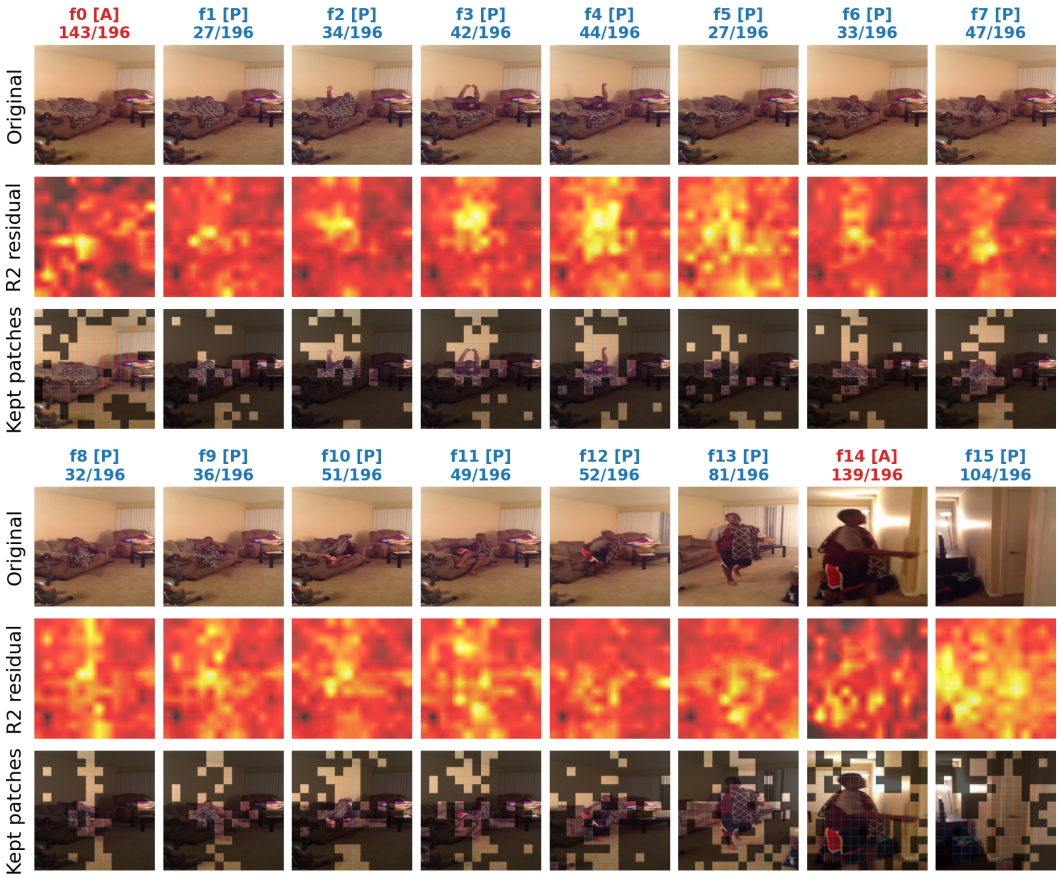

Figure 5: Qualitative PreViC token selection on one 16-frame video at $\alpha = 0.3$, shown as two blocks of eight frames. Each block shows the original frames, the R2 predictive-residual map and the kept patches, one row each. Red labels mark the two R1 anchor frames ([A]) and blue labels the predicted non-anchor frames ([P]), and each label gives the number of tokens kept in that frame out of 196.

D Time Complexity and Implementation Details

Scoring-space and regularization details. Equation 2 should be interpreted as a local predictability model in the post-projector visual-token space, not as an assumption that LLM hidden states are globally linear. This follows the intuition of local linear reconstruction (Roweis & Saul, 2000). Nearby tokens provide a compact basis, and the residual measures the component unexplained by that local neighborhood. This view is also consistent with prior analyses showing meaningful linear structure in multimodal representation spaces (Bhalla et al., 2024).

We use cosine similarity for scale-insensitive neighbor retrieval, while measuring the residual in the post-projector feature space consumed by the LLM. Cosine similarity retrieves semantically similar anchors regardless of token norm, whereas the L2 residual preserves both directional and magnitude differences that the anchor basis cannot explain. The implementation measures the residual on raw post-projector features.

We choose the ridge coefficient adaptively for numerical stability:

$$\lambda = \max\left(\epsilon,\ \epsilon \cdot \frac{\mathrm{tr}(\mathbf{B}_\mathbf{x}\mathbf{B}_\mathbf{x}^\top)}{k_{\mathrm{nn}}}\right), \quad \epsilon = 10^{-3}. \tag{4}$$

This adaptive choice scales the regularization with the average squared norm of retrieved anchor tokens and stabilizes the solution when nearest anchors are highly correlated.

The cost of anchor selection is $O(FTD)$. With $N_{\text{anc}} = aT$, exact cosine search over the anchor pool for the N_{res} non-anchor targets costs $O(N_{\text{res}} N_{\text{anc}} D)$. Constructing each local $k_{\text{nn}} \times k_{\text{nn}}$ Gram matrix costs $O(k_{\text{nn}}^2 D)$ per target and its dense solve costs $O(k_{\text{nn}}^3)$. Reconstruction adds $O(k_{\text{nn}} D)$ per target. The total worst-case preprocessing cost is therefore $O(FTD + N_{\text{res}}[N_{\text{anc}} D + k_{\text{nn}}^2 D + k_{\text{nn}}^3])$, excluding the upstream vision encoder and projector. The fixed implementation batches the cosine search and the ridge solves on GPU, forming and solving each system in float32 with a pseudo-inverse fallback, while the selected embeddings keep the model input's dtype. Small default values of a and k_{nn} do not remove the anchor-search dependence on input length. The observed compressor time and end-to-end effect must be measured for the target backbone and video length.