STORS: SPATIO-TEMPORAL ORTHOGONAL RESIDUAL SELECTION FOR VIDEO TOKEN COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient video understanding requires compressing highly redundant visual tokens across space and time. Existing geometric selection methods typically operate on decoder-facing representations, overlooking how the visual projector may alter the geometry on which their selection criteria depend. We identify a substantial change in representation geometry across the projector in Qwen2.5-VL: encoder-side features exhibit extreme anisotropy and near-saturated pairwise similarities, whereas decoder-facing features are substantially more dispersed. Motivated by this finding, we propose STORS, a training-free method that scores tokens by their unexplained residual energy over a unified spatio-temporal pool while retaining their projected features for the language model. An explicit temporal redundancy weight controls cross-frame redundancy under a fixed token budget. Controlled experiments reveal that the effectiveness of residual selection depends strongly on the scoring space: on Video-MME, the same criterion underperforms uniform grid sampling in all nine settings when applied after the projector, but outperforms it in eight of nine when applied before the projector. In contrast, no statistically detectable scoring-space benefit is observed in LLaVA-OneVision, whose representations exhibit similar geometry. Across five video benchmarks and two backbones, STORS achieves competitive performance under aggressive compression. On Qwen2.5-VL-7B, retaining 10% of visual tokens preserves 94.4% of the uncompressed average accuracy while achieving a $1.82\times$ speedup in time to first token.

1 INTRODUCTION

Vision-Language Models (VLMs) have recently achieved remarkable progress on visual understanding and multimodal reasoning tasks (Li et al., 2023; Zhang et al., 2023; Liu et al., 2024; Li et al., 2025; Bai et al., 2025). When the input is extended from a single image to a video, the number of visual tokens grows linearly with the number of frames. Since video VLMs typically encode each frame independently, a 32-frame video already yields more than 8,000 visual tokens, whereas the textual side usually contains only a handful of tokens. As a result, visual tokens not only occupy the vast majority of the context length during video reasoning, but also dominate the computation of the prefill stage and the GPU memory consumed by the KV cache, as illustrated in Figure 1. As videos become longer, this issue is especially pronounced in scenarios where the input may span several minutes or even hours, so that the limited GPU memory and context window quickly become the bottleneck of deployment.

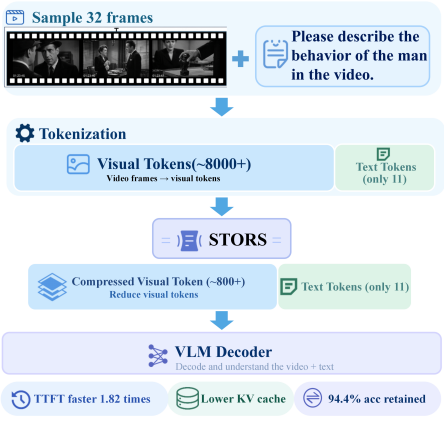


Figure 1: The Visual Token Bottleneck. 32 sampled frames yield over 8,000 visual tokens, compared with just 11 text tokens. At a 10% retention ratio, STORS retains 94.4% of the uncompressed accuracy while achieving a $1.82\times$ speedup in TTFT and reducing KV-cache usage.

Videos exhibit substantial spatio-temporal redundancy: adjacent frames usually depict continuous visual states,

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.

in which a large amount of information is repeated over
time, while different regions within a single frame often carry correlated or duplicated spatial information. Once such video content is encoded into visual tokens, this spatio-temporal redundancy persists. It is therefore possible to reduce repeated information by compressing visual tokens, lowering the token count while preserving as much of the effective video information as possible, and thereby alleviating the pressure on the context length, the computational cost and the KV cache memory of video VLMs. Since retraining a video VLM is expensive, training-free and plug-and-play token compression constitutes a directly applicable solution.

Existing training-free methods are grouped into three categories. Spatial methods such as VisionZip (Yang et al., 2025) and PruMerge (Shang et al., 2025) rank tokens by attention weights or embedding similarity within a frame. Temporal methods such as DyCoke (Tao et al., 2025) and TempMe (Shen et al., 2025b) merge similar tokens across neighbouring frames. In addition, staged spatio-temporal methods such as PruneVID (Huang et al., 2025) and FastVID (Shen et al., 2025a) first group frames into segments and then reduce redundancy inside each segment, whereas FlashVID (Fan et al., 2026) merges tokens along similarity-based spatio-temporal trees grown across successive frames.

Two limitations remain in existing methods. First, geometric selection criteria typically operate on decoder-facing features, without considering how the projector changes token geometry. Such changes can alter the spectral structure and pairwise similarities of visual tokens, making the scoring space itself a critical design choice rather than a fixed property of the model. Second, existing methods often restrict redundancy reduction to individual frames or predefined temporal segments. This limits global token allocation and leaves the relative importance of within-frame and cross-frame redundancy implicit. Attention-based methods also depend on attention weights, which may be unavailable with fused attention kernels such as FlashAttention (Dao et al., 2022).

Motivated by these limitations, we propose STORS (Spatio-Temporal Orthogonal Residual Selection), a training-free method for video token compression. We first examine the geometry of visual tokens before and after the projector and find that the two representations can differ substantially in models such as Qwen2.5-VL. Encoder-side features are highly anisotropic and exhibit near-saturated pairwise cosine similarities, while decoder-facing features are much more dispersed. This makes raw similarity less discriminative in the encoder space, whereas orthogonal residuals can still measure how much information remains unexplained by the selected set. STORS therefore separates the space used for scoring from the features passed to the language model: tokens are scored before the projector, while their projected features are retained for decoding. To exploit redundancy across both space and time, STORS places all video tokens in a unified spatio-temporal pool and greedily selects those with the largest unexplained residual energy. A temporal redundancy weight further controls the contribution of cross-frame redundancy, enabling global token allocation without predefined temporal segments.

We summarize our contributions as follows:

- We identify a substantial geometric difference across the projector in Qwen2.5-VL. Encoder-side features exhibit strong anisotropy and near-saturated pairwise similarities, while decoder-facing features are more dispersed. We show that this difference leads to substantially different selection performance and motivates separating the scoring space from the representation retained for the language model.
- We propose STORS, a training-free and attention-free method that performs orthogonal residual selection over a unified spatio-temporal token pool. It scores tokens by their unexplained residual energy and uses an explicit temporal redundancy weight to control cross-frame redundancy under a strictly conserved token budget.
- Experiments on five video benchmarks and two backbones show that the effectiveness of residual selection depends on the representation in which it is applied. The scoring-space advantage is substantial on Qwen2.5-VL's Video-MME, but is not statistically detectable on LLaVA-OneVision, where the two representations exhibit similar geometry. STORS remains competitive under aggressive compression and matched token budgets.

2 Related Work

Training-free token compression. Training-free methods reduce visual tokens at inference without updating parameters (Shao et al., 2025b). They differ in the signal that decides what is kept. ToMe (Bolya et al., 2023) merges tokens inside the vision transformer by feature similarity. Similarity finds duplicates, not usefulness. A second group therefore uses attention. FastV (Chen et al., 2024) prunes visual tokens in deep decoder layers, where they receive little attention; VisionZip (Yang et al., 2025) applies a similar criterion before the decoder and merges the remainder into a few contextual tokens. PruMerge (Shang et al., 2025), PyramidDrop (Xing et al., 2025) and SparseVLM (Zhang et al., 2025b) vary the reduction site and whether removed tokens are recycled. Since attention concentrates on few regions, top-ranked tokens are mutually redundant. Subset-level criteria address this: DivPrune (Alvar et al., 2025) maximises the minimum pairwise distance, DART (Wen et al., 2025) expands from pivots to dissimilar neighbours, CDPruner (Zhang et al., 2025a) maximises a query-conditioned DPP, and MMTok (Dong et al., 2026) maximises submodular coverage. In all cases the decisive quantity is an attention weight or a cosine similarity, read from the decoder-facing features.

Video token compression. Applied per frame, such criteria miss the dominant redundancy of video: consecutive frames repeat one another. DyCoke (Tao et al., 2025) merges tokens across adjacent frames and prunes the KV cache, but only within a local window. Segment-based methods widen the scope. PruneVID (Huang et al., 2025) clusters frames into segments and separates static from dynamic regions; FastVID (Shen et al., 2025a) makes the segmentation dynamic; HoliTom (Shao et al., 2025a) uses a global dynamic-programming segmentation plus in-decoder merging. Others vary the grouping granularity, from clip-level merging (Shen et al., 2025b) and joint aggregation (Ren et al., 2023) to semantic components (Sun et al., 2025) and tree-structured merging (Fan et al., 2026). All share a factorised design: grouping precedes reduction. The grouping thus fixes in advance which redundancy is removable, and the balance between temporal and spatial redundancy stays implicit in a heuristic.

Objective-driven selection. A separate line replaces heuristics with an explicit objective: keep the subset that best reconstructs the discarded tokens. SPARE (Lee et al., 2026) casts pruning as column subset selection, solves it by a greedy rank-revealing QR sweep, and bounds the output perturbation by the reconstruction error. CDPruner (Zhang et al., 2025a) reads the same sweep as orthogonal matching pursuit and links it to volume sampling through the determinant recursion of a determinantal point process. ToDRE (Li et al., 2026) combines a diversity stage with a relevance stage. These objectives are posed per image; even where they are applied to video, redundancy from content persisting across frames has no counterpart in them, and selection and representation share one feature space.

Unified allocation. Du et al. (2026) question separability itself. Staged decisions misallocate the budget at low retention, keeping inessential tokens in some frames while discarding critical ones in others. They pool all frames into one candidate set under a single global constraint. Selection maximises informational contribution and penalises semantic redundancy; unselected tokens are clustered by DPC-KNN and reinserted, and a query-guided stage merges again inside the decoder. The unification concerns the pool over which selection ranges. The ranking quantity remains an attention weight and a cosine similarity from decoder-facing features.

3 Our Proposed Method

Token compression asks which visual tokens survive. We take the position that the kept set should be the one from which the discarded ones are best reconstructed: it turns an intractable information objective into a subset selection problem with a greedy solution whose every quantity is an orthogonal residual. This reconstruction view, however, is silent on two points as soon as the input is a video rather than an image. It never asks *which* kept tokens explain a candidate, so a video is treated as a longer image; and it never says in *which* representation the residual should be measured, a free variable that prior work fixes by default rather than by argument. Our method, STORS (Spatio-Temporal Orthogonal Residual Selection), addresses both questions. We detail the resulting procedure below, as sketched in Figure 2.

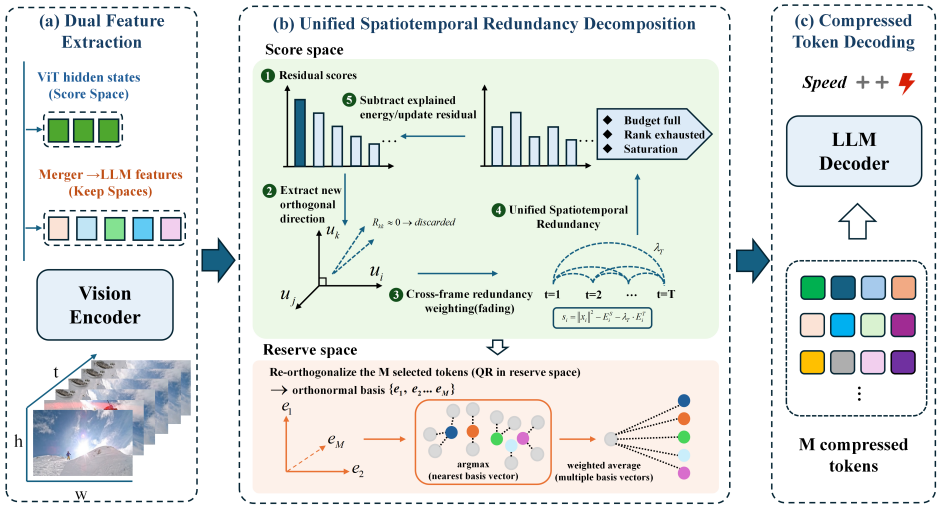


Figure 2: Overview of STORS. (a) The encoder yields two representations of the same token index set: the pre-projector hidden states (*score space*) and the post-projector features (*reserve space*). (b) Selection runs entirely in the score space, as a block-greedy orthogonal residual decomposition over a single spatio-temporal candidate pool in which explained energy coming from other time slices is discounted by the temporal redundancy weight λ_T. (c) The selected tokens enter the decoder with their original positional encodings preserved, thereby accelerating the decoding process.

3.1 Problem Formulation

Setup. A video LLM consists of a vision encoder, a projector and an LLM decoder. A video is sampled into T time slices of N spatial positions each, giving $L = TN$ visual tokens $V = \{v_1, \ldots, v_L\}$; we write $t(i)$ for the slice of token i. The same index set carries two representations: the encoder hidden states $Z \in \mathbb{R}^{L\times d'}$ produced before the projector, and the features $X = [x_1, \ldots, x_L]^\top \in \mathbb{R}^{L\times d}$ produced after it, which are what the decoder consumes. Given a retention ratio ρ, compression selects $S \subseteq V$ with $|S| = M = \mathrm{round}(\rho L)$ and forwards only the corresponding features. The budget is conserved exactly rather than on average, so two methods compared at the same ρ receive an identical number of tokens.

From information to reconstruction. The criterion we would like to optimise is the subset that is maximally informative about the whole set, $\arg\max_{|S|=M} I(V;S)$, which is not computable. The Shannon lower bound on the minimum reconstruction error $\xi(S)$ of all tokens from S gives $I(V;S) \geq H(V) - \frac{d}{2}\log(2\pi e\,\xi(S))$, and since $H(V)$ does not depend on S, minimising reconstruction error maximises a computable lower bound on the mutual information. Restricting the reconstruction map to linear ones – which avoids training an auxiliary decoder and only loosens the bound, leaving the direction of optimisation unchanged – replaces ξ by an orthogonal projection onto the span of the kept tokens.

$$S^\star = \arg\min_{S\subseteq V,\ |S|=M} J(S), \qquad J(S) = \sum_{i=1}^{L} \left\| x_i - P_{\mathrm{span}(S)}\, x_i \right\|_2^2. \tag{1}$$

The practical reading of Eq. (1) is that a token matters exactly insofar as it cannot be rebuilt from the tokens already kept, which is a statement about the geometry of the features alone and involves no attention weights.

Greedy orthogonal residual. Minimising Eq. (1) exactly is NP-hard; the standard tractable surrogate is a column-pivoted QR sweep that maintains an orthonormal basis $\{q_1, \ldots, q_m\}$ of the tokens

selected so far and repeatedly takes the token with the largest residual outside that span,

$$r_i \;=\; \|x_i\|^2 - \sum_{k \le m} \langle x_i, q_k \rangle^2. \tag{2}$$

Each marginal gain is thus readable directly from the running basis, and rank-revealing QR bounds the resulting subset against the best rank-M approximation.

3.2 Pricing Temporal Redundancy

Eq. (2) asks how much of a token is still unexplained but never by whom. For a video this erases the only structure that matters. Redundancy there has two sources of different character: within a slice it arises from the continuity of local texture, across slices from the persistence of the scene, and the second can reach across arbitrary temporal distance while the first cannot. A criterion that sums both into one number can express neither the preference for new content over repeated content, nor its opposite.

We therefore split the explained energy of a candidate by the slice of the basis direction that explains it. Writing $E_i^S = \sum_{k:\, t(q_k)=t(i)} \langle x_i, q_k \rangle^2$ for the energy contributed by already selected tokens in the *same* slice as i and $E_i^T = \sum_{k:\, t(q_k)\neq t(i)} \langle x_i, q_k \rangle^2$ for that contributed by *other* slices, so that their sum is exactly the subtracted term of Eq. (2), we rank candidates by

$$s_i \;=\; \|x_i\|^2 - E_i^S - \lambda_T\, E_i^T, \qquad \lambda_T \in [0, 1]. \tag{3}$$

The weight λ_T is an exchange rate between temporal and spatial redundancy, and the two ends of its range are two familiar regimes. At $\lambda_T = 1$ a cross-slice explanation counts as fully as a within-slice one: once content has been selected in *any* slice, its counterparts in *all* other slices are suppressed at once and the budget flows towards content that has not been seen; here $s_i \equiv r_i$ and the criterion is plain column subset selection over one global pool. At $\lambda_T = 0$ cross-slice explanation is not redundancy at all, and static content is re-selected in every frame. Restricting in addition the basis scope to a single slice recovers frame-wise orthogonal residual selection token for token, so per-frame methods are a strict special case of Eq. (3) rather than a separate family, and λ_T parameterises the whole path between the two regimes inside one implementation.

The score is not the residual. Subtracting Eq. (2) from Eq. (3) gives

$$s_i - r_i \;=\; (1 - \lambda_T)\, E_i^T \;\ge\; 0 \qquad (\lambda_T \le 1), \tag{4}$$

so for $\lambda_T < 1$ the ranking quantity is no longer the true residual. A token that is almost entirely explained from other slices can then simultaneously carry the highest score and a residual near zero, and a saturation test written on s would read that as convergence and abandon the greedy pass in its first rounds, silently handing the budget to whatever fallback fills the remaining slots. We therefore rank with s and test termination with r.

3.3 Score and Reserve Spaces

A residual is defined only relative to a representation. The criterion developed so far prescribes how tokens are compared, but not the features on which they are compared. The bound behind Eq. (1) is indifferent to that choice and holds for any intermediate representation of the same tokens. A video VLM offers two of them. The encoder produces hidden states Z before the projector, and the projector produces the features X that the decoder consumes. Prior work scores on X because the decoder receives X. The derivation does not require it. We therefore separate what we score on from what we keep, and call Z the *score space* and X the *reserve space*.

$$S = \mathrm{STORS}\big(Z;\, \rho, \lambda_T\big), \qquad \text{decoder input} = X_S, \tag{5}$$

Which representation serves as Z is a property of the model, not of the criterion. A projector that merges spatially adjacent tokens through a learned transform leaves the two spaces geometrically unrelated. The linear structure Eq. (1) operates on is then preserved only before the projector. We accordingly take Z to be the pre-projector encoder states.

Algorithm 1 STORS selection (block-greedy, single spatio-temporal pool)

Input: Z (score space, slice index $t(i)$), X (reserve space), budget M, weight λ_T, block size b, tolerance ϵ, strength μ
$r_i \leftarrow \|z_i\|^2,\ s_i \leftarrow \|z_i\|^2,\ S \leftarrow \varnothing,\ Q \leftarrow \varnothing$
while $|S| < M$, the rank is not exhausted, and $\max_i r_i/\|z_i\|^2 \geq \epsilon$ **do**
  $B \leftarrow$ top-b unselected by s; $\{q_k\} \leftarrow \mathrm{QR}\big((I - QQ^\top)Z_B^\top\big)$; $S \leftarrow S \cup B$; $Q \leftarrow [\,Q, \{q_k\}\,]$ {near-collinear candidates are dropped}
  $r_i \mathrel{-}= \langle z_i, q_k\rangle^2,\ s_i \mathrel{-}= \langle z_i, q_k\rangle^2 \cdot \big(1 \text{ if } t(q_k) = t(i) \text{ else } \lambda_T\big)$ for all i and new q_k
end while
fill the remaining $M - |S|$ slots by phase-rotating spatio-temporal grid sampling
$Q_X \leftarrow \mathrm{QR}(X_S^\top);\ c_{ij} \leftarrow \cos\big(Q_X x_i, Q_X x_j\big);\ a_i \leftarrow \max_{j\in S} c_{ij},\ \pi(i) \leftarrow \arg\max_{j\in S} c_{ij}$ for $i \notin S$
$\tilde{x}_j \leftarrow \big(x_j + \mu\sum_{\pi(i)=j} a_i x_i\big)\big/\big(1 + \mu\sum_{\pi(i)=j} a_i\big)$ for $j \in S$
Output: S with $|S| = M$; the decoder receives $\tilde{X}_S$

Only the index set crosses the boundary. Sequence length, token content and positional semantics remain those of the uncompressed pipeline restricted to S. Two consequences follow for the implementation. A basis built in Z is no coordinate system in X, so redistributing content among the kept tokens requires re-orthogonalising them in X. The rank available to the sweep is that of the score space, so $M \leq d'$.

3.4 Unified Spatio-Temporal Pool Selection

Algorithm 1 instantiates the two decisions above. Selection runs as a single block-greedy pass over one spatio-temporal candidate pool judged against one basis for the whole video, so no time-slice boundary stands in the way of removing redundancy. Each round orthogonalises the current top-b candidates against the running basis and factorises them, which does double duty: it strips the components that are already explained, and it suppresses near-duplicates inside the block, since nearly collinear candidates produce a vanishing diagonal entry and are dropped. The loop ends when the budget is filled, when the rank of the score space is exhausted, or upon saturation, the test that by Eq. (4) must be written on r; whatever budget remains is filled by a phase-rotating spatio-temporal grid sampler. Discarded tokens are not thrown away. We re-orthogonalise the kept set in the reserve space, which makes the coordinates comparable across the whole segment, and return each discarded token to the kept token closest to it there, weighted by that closeness and scaled by a merging strength μ. Merging never changes the token count. It only changes the content of the tokens that are already being kept.

4 Experiments

4.1 Setup

Models and benchmarks. We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025). As a cross-backbone we conduct experiments on LLaVA-OneVision-7B (Li et al., 2025), whose projector acts on each token independently. We adopt five popular datasets. Video-MME (Fu et al., 2025) involves short, medium and long splits, independently. MVBench (Li et al., 2024) includes 20 tasks. EgoSchema (Mangalam et al., 2023) has the local subset and the full split scored by the official server. MLVU (Zhou et al., 2025) includes multiple-choice tasks with M-Avg score and two generative tasks with G-Avg score. In addition, VideoChatGPT (Maaz et al., 2024) covers five dimensions: Correctness of Information, Detail Orientation, Contextual Understanding, Temporal Understanding, and Consistency. In every benchmark, all videos are uniformly sampled to 32 frames.

Baselines and metrics. All methods are compared under an equal visual token budget, defined as a fixed retention ratio $\rho \in \{5\%, 10\%, 20\%\}$ of the visual tokens the uncompressed model would process. All experiments are run on a single NVIDIA RTX PRO 6000 GPU, with the same decoding configuration for every method. We compare against VisionZip (Yang et al., 2025), FastVID (Shen et al., 2025a), FlashVID (Fan et al., 2026), DyCoke (Tao et al., 2025), PruneVID (Huang et al., 2025) and the uncompressed model. Unless stated otherwise, STORS runs in the score space with $\lambda_T = 1$ and merging strength $\mu = 0.5$. Multiple-choice accuracy is reported for Video-MME, MVBench,

Table 1: Qwen2.5-VL-7B on video understanding benchmarks.

Method	Retention Ratio ρ	Video-MME			MLVU	EgoSchema		MVBench	Avg.	
		Short	Medium	Long		Subset	Full		Score	Rel. Acc (%)
Vanilla	100%	72.6	60.1	52.1	61.3	63.6	59.2	66.6	62.2	100.0
VisionZip (Yang et al., 2025)	20%	70.2	56.1	51.1	58.2	62.4	56.9	**63.7**	59.8	96.1
FastVID (Shen et al., 2025a)		69.9	55.8	49.7	57.8	57.2	56.4	62.3	58.4	93.9
FlashVID (Fan et al., 2026)		67.2	55.6	47.9	59.3	**63.6**	**57.2**	63.1	59.1	95.0
DyCoke (Tao et al., 2025)		69.2	55.3	50.7	58.1	60.4	56.6	61.8	58.9	94.6
PruneVID (Huang et al., 2025)		70.1	**56.8**	50.4	58.7	60.8	56.3	62.0	59.3	95.3
STORS (ours)		**71.2**	56.3	**51.4**	**60.2**	61.0	56.8	63.1	**60.0**	**96.4**
VisionZip (Yang et al., 2025)	10%	67.1	54.2	50.3	56.5	57.4	54.8	**61.8**	57.4	92.3
FastVID (Shen et al., 2025a)		66.3	54.2	49.0	55.2	56.0	55.6	59.3	56.5	90.8
FlashVID (Fan et al., 2026)		64.6	54.2	47.8	57.7	59.8	**55.9**	61.5	57.4	92.2
DyCoke (Tao et al., 2025)		65.8	53.8	48.6	55.9	57.0	54.7	59.5	56.5	90.8
PruneVID (Huang et al., 2025)		66.7	54.5	49.8	56.1	57.8	54.6	59.7	57.0	91.7
STORS (ours)		**69.3**	**55.4**	**51.6**	**58.4**	**60.2**	55.1	61.2	**58.7**	**94.4**
VisionZip (Yang et al., 2025)	5%	62.6	**52.7**	47.9	54.2	54.0	52.4	**58.7**	54.6	87.8
FastVID (Shen et al., 2025a)		61.4	51.1	45.9	52.3	56.9	51.7	56.2	53.6	86.2
FlashVID (Fan et al., 2026)		61.8	50.8	45.6	54.5	**57.4**	52.1	55.8	54.0	86.8
DyCoke (Tao et al., 2025)		62.1	51.7	46.8	50.1	56.8	51.9	56.6	53.7	86.3
PruneVID (Huang et al., 2025)		63.0	51.9	47.2	51.2	57.1	51.2	56.9	54.1	86.9
STORS (ours)		**66.3**	52.6	**50.0**	**55.8**	56.2	**52.7**	58.6	**56.0**	**90.1**

Table 2: LLaVA-OneVision-7B, average score per benchmark.

Dataset	Vanilla	Retention Ratio $\rho = 20\%$					Retention Ratio $\rho = 10\%$				
	(100%)	FastV	VisionZip	FastVID	FlashVID	**Ours**	FastV	VisionZip	FastVID	FlashVID	**Ours**
Video-MME	59.3	55.7	58.0	57.9	58.2	**58.6**	52.7	53.3	55.2	55.8	**56.4**
EgoSchema	62.7	59.1	61.0	60.4	61.2	**62.3**	57.5	60.1	58.8	60.5	**62.1**
MVBench	57.8	56.0	56.6	57.1	**57.2**	56.6	53.4	54.8	56.0	**56.4**	55.4
Avg. Score	59.9	56.9	58.5	58.5	58.9	**59.2**	54.5	56.1	56.7	57.6	**58.0**
Rel. Acc (%)	100.0	95.0	97.7	97.6	98.2	**98.8**	91.0	93.5	94.5	96.1	**96.7**

EgoSchema and MLVU, while generative quality on VideoChatGPT and MLVU(gen) is scored by an LLM judge.

4.2 Main Results

Results on Qwen2.5-VL in Table 1 report the relative accuracy under three retention ratios.STORS ranks first at every budget, and its margin widens as the budget tightens. At $\rho = 20\%$ all methods are close. The gap appears at 10% and grows at 5%, where every baseline drops below 88% of the uncompressed model while STORS holds near 90%. The gains are not uniform across benchmarks. Their pattern tracks how much temporal redundancy each benchmark offers. The largest gains occur on long-form settings, namely the long split of Video-MME and MLVU. Content there persists across many frames, so a criterion that suppresses a region everywhere once it is selected somewhere has more to exploit. On MLVU the segment-based baselines degrade sharply at the tightest budget, consistent with segments that become too coarse to allocate a very small budget. The smallest gains occur on MVBench, where STORS is not the leader. Its clips are short, and its questions hinge on fine-grained motion and attribute distinctions. Cross-frame redundancy is thus limited, and attention-based salience stays slightly ahead.

We further evaluate our method on LLaVA-OneVision-7B (Table 2), where it consistently achieves the best average score and relative accuracy among all compared methods at both retention ratios, with the gap to the vanilla model remaining marginal even under aggressive compression. Across individual benchmarks, Ours performs particularly well on Video-MME and EgoSchema, while remaining competitive on MVBench despite occasionally trailing FlashVID by a small margin. This stable, benchmark-agnostic behavior as the retention ratio decreases indicates that our selection strategy reliably retains semantically important visual tokens, confirming the effectiveness and robustness of the proposed approach in preserving multimodal reasoning capability under substantial token reduction.

Table 3: Open-ended generation uses Qwen2.5-VL-7B and is rated 1–5 by an LLM judge. We evaluate on the VideoChatGPT and MLVU tasks.

Metric	Vanilla	$\rho = 5\%$			$\rho = 10\%$			$\rho = 20\%$		
	(100%)	VisionZip	FlashVID	**Ours**	VisionZip	FlashVID	**Ours**	VisionZip	FlashVID	**Ours**
Correctness	3.259	3.122	3.185	**3.281**	3.184	3.268	**3.317**	3.270	3.300	**3.338**
Detail	3.005	2.948	3.047	**3.124**	2.986	3.077	**3.138**	3.034	3.083	**3.145**
Context	3.534	3.415	3.508	**3.634**	3.488	3.595	**3.652**	3.555	3.611	**3.654**
Temporal	2.946	2.781	2.870	**2.887**	2.831	2.930	**2.939**	2.932	2.954	**2.972**
Consistency	3.445	3.289	3.411	**3.483**	3.369	3.475	**3.505**	3.445	3.461	**3.529**
MLVU (Sub-Scene)	2.31	2.090	2.105	**2.174**	2.045	2.100	**2.144**	2.070	**2.090**	2.070
MLVU (Summary)	2.25	1.922	**1.963**	1.917	**2.009**	1.995	1.963	2.009	2.042	**2.046**

To evaluate our method more comprehensively, we further test it on open-ended generation tasks using the Qwen2.5-VL-7B model. Table 3 reports results on open-ended generation. STORS outperforms both baselines on every VideoChatGPT metric at every budget. The pattern across metrics is informative. The largest margins appear on detail and context, the ratings that reward covering what the video contains rather than locating one salient region. The smallest margins appear on the temporal rating, where all methods lose the most. Ordering and change require evidence from frames that a tight budget cannot retain, whatever the selection rule. Scores also move little with the budget: VisionZip degrades steeply from $\rho = 20\%$ to 5%, while STORS stays nearly flat. The two MLVU generative tasks form a different regime. Every method loses a substantial fraction of the uncompressed rating, far more than on any multiple-choice benchmark. The remaining differences are small and not monotone in the budget. Summarising an hour-long video requires evidence spread thinly over the sequence rather than concentrated in a few regions. Deciding *which* tokens to keep matters less when the answer needs more than the budget allows.

4.3 Ablations and Analysis

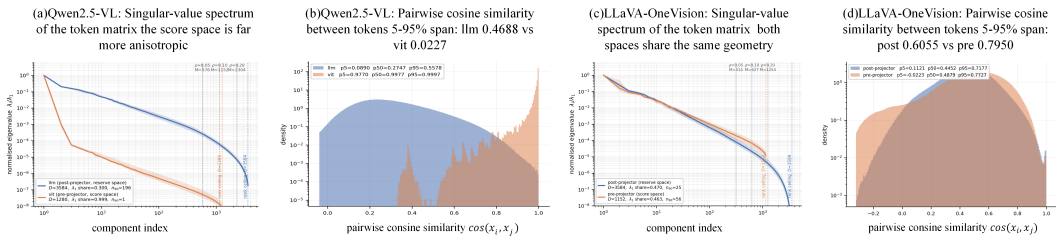


Figure 3: Geometry of the score and reserve spaces under two projector designs. (a,b) Qwen2.5-VL, whose merger applies a learned MLP to four spatially adjacent sub-tokens; (c,d) LLaVA-OneVision, whose projector acts per token. (a,c) Normalised singular-value spectrum of the token matrix; dotted lines mark each space's rank ceiling D, dashed lines the budgets at $\rho \in \{5\%, 10\%, 20\%\}$. (b,d) Distribution of pairwise cosine similarity between tokens. The two spaces are geometrically unrelated on Qwen2.5-VL and nearly coincide on LLaVA-OneVision. Curves are medians over seven clips; shaded bands give the 10–90% range.

Table 4: Effect of the scoring space across datasets and backbones.

Dataset (split)	$\rho = 5\%$					$\rho = 10\%$					$\rho = 20\%$				
	reserve	score	Δ	b/c	p	reserve	score	Δ	b/c	p	reserve	score	Δ	b/c	p
Qwen2.5-VL-7B															
Video-MME (Short)	60.78	**66.33**	+5.56	44/94	3×10^{-5}	63.67	**69.33**	+5.67	34/85	$< 10^{-5}$	63.67	**71.22**	+7.56	27/95	$< 10^{-5}$
Video-MME (Medium)	50.22	**52.56**	+2.33	50/71	0.069	51.56	**55.44**	+3.89	49/84	0.003	53.00	**56.33**	+3.33	41/71	0.006
Video-MME (Long)	45.78	**50.00**	+4.22	35/73	0.0003	47.22	**51.56**	+4.33	36/75	0.0003	48.78	**51.44**	+2.67	34/58	0.016
EgoSchema (Subset)	55.4	**56.2**	+0.80	17/21	0.627	60.0	**60.2**	+0.20	21/22	1.000	**62.0**	61.0	−1.00	18/13	0.473
MVBench	**59.00**	58.55	−0.45	217/199	0.405	**61.50**	61.23	−0.27	198/187	0.610	**63.68**	63.10	−0.58	194/172	0.272
LLaVA-OneVision-7B															
Video-MME (Short)	**66.00**	65.67	−0.33	36/33	0.810	66.89	**67.11**	+0.22	28/30	0.896	**71.11**	70.56	−0.56	37/32	0.631
Video-MME (Medium)	51.62	**52.00**	+0.38	41/46	0.668	**54.62**	54.22	−0.40	41/38	0.822	55.28	**55.56**	+0.28	39/42	0.824
Video-MME (Long)	**47.22**	46.49	−0.73	36/28	0.382	**48.11**	47.89	−0.22	34/32	0.902	49.56	**49.78**	+0.22	29/31	0.897
EgoSchema (Subset)	**65.0**	64.0	−1.00	18/13	0.473	64.8	**65.2**	+0.40	15/17	0.860	63.4	**63.6**	+0.20	6/7	1.000
MVBench	**54.17**	53.37	−0.80	173/140	0.070	55.00	**55.40**	+0.40	135/143	0.675	56.50	**56.60**	+0.10	109/113	0.841

We further analyse the representation from which the residual is read. Table 4 varies this single factor. On Qwen2.5-VL the encoder space wins on Video-MME. The contrast between the two backbones is explained by the projector. Figure 3 compares the two spaces on each backbone. Qwen's merger applies a learned MLP to four spatially adjacent sub-tokens, and its two spaces are geometrically unrelated. The encoder spectrum collapses onto a single dominant direction, while the post-projector spectrum spreads over hundreds, and the two similarity distributions hardly overlap. LLaVA's projector instead transforms each token independently and preserves the relative geometry. The two spectra nearly coincide and the similarity distributions largely overlap, so both spaces induce almost the same ordering and return almost the same subset. The size of the scoring-space effect is thus bounded by how far the projector deforms the geometry it is given, and mixing spatially adjacent tokens is what deforms it.

Table 5: Inference efficiency at ρ=10% on Video-MME; 32 (top), 64 (bottom) frames.

Method	KV (MB)	TTFT (s)	Speed up	Extra (GB)
No compression	639.2	1.10	1.00×	-
VisionZip	72.2	0.64	1.73×	2.1
STORS	72.2	**0.60**	**1.82**×	**2.0**
FlashVID	72.4	0.71	1.53×	21.0
No compression	1269.2	2.34	1.00×	-
VisionZip	135.2	1.24	1.88×	4.2
STORS	135.2	**1.18**	**1.99**×	**3.9**
FlashVID	135.7	1.38	1.70×	41.9

Table 6: Ablation study of the merging strength μ on Video-MME with Qwen2.5-VL-7B.

Configuration	$\rho = 10\%$	$\rho = 20\%$
reserve space, $\mu = 0$	53.15	55.67
reserve space, $\mu = 0.5$	54.15	55.15
score space, $\mu = 0$	58.33	58.93
score space, $\mu = 0.5$	**58.78**	**59.67**
score space, $\lambda_T = 0$	56.63	58.13
score space, $\lambda_T = 0.25$	56.70	58.67
score space, $\lambda_T = 0.5$	56.59	58.37
score space, $\lambda_T = 0.75$	56.74	58.41
score space, $\lambda_T = 1$	**58.77**	**59.63**

We compare STORS with two representative methods, VisionZip and FlashVID, on Qwen2.5-VL-7B, under a retention ratio of $\rho = 10\%$, as shown in Table 5. STORS achieves the largest speedup in time-to-first-token at both 32 and 64 frames ($1.82\times$/$1.99\times$, outperforming both VisionZip and FlashVID), while incurring the smallest additional memory overhead.

Table 6 reports the effect of the scoring space, the merging strength μ, and λ_T under the same split. The choice of scoring space proves decisive: moving the criterion into the encoder (score) space improves accuracy at both budgets. Adjusting μ within a fixed space, by contrast, has a comparatively small effect. Over the intermediate range of λ_T, accuracy remains essentially flat, showing no clear trend as the value increases; only with full temporal generalization does the result move out of this plateau, and it does so consistently at both budgets. Partial extension of the temporal range therefore behaves much like a purely local criterion, and the advantage only emerges once the criterion is allowed to cover the entire segment. We accordingly adopt $\lambda_T = 1$ and $\mu = 0.5$ as the default configuration of our method.

5 Conclusion

We presented STORS, a training-free method for video token compression. Through experiments, we found that the aligned encoder-side and decoder-facing representations can exhibit substantially different geometry, and that the effectiveness of residual-based selection depends on the representation used for scoring. Motivated by this observation, STORS separates the score space used for selection from the reserve space that supplies features to the language model. It performs orthogonal residual selection over a unified spatio-temporal pool, with a temporal redundancy weight controlling the contribution of cross-slice explained energy. On two backbones with different projector designs, STORS achieves competitive performance across several video benchmarks under matched token budgets, with larger average gains at lower retention ratios. The benefit of the scoring-space choice, however, remains dependent on the backbone and task. A remaining limitation arises on questions requiring specific evidence, such as counting actions, ordering events, and determining motion direction. A subset chosen to span the visual features is not necessarily the subset that preserves the evidence such questions need. Extending the reconstruction view to account for what a question asks of the retained set is a natural next step.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, 2025. arXiv:2503.02175.

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

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. Oral. arXiv:2210.09461.

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 (ECCV)*, pp. 19–35, 2024. Oral. arXiv:2403.06764.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. MMTok: Multimodal coverage maximization for efficient inference of VLMs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. arXiv:2508.18264.

Junhao Du, Jialong Xue, Anqi Li, Jincheng Dai, and Guo Lu. Unified spatiotemporal token compression for video-LLMs at ultra-low retention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2603.21957.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. FlashVID: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. Oral. arXiv:2602.08024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, 2025. arXiv:2405.21075.

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. doi: 10.18653/v1/2025.findings-acl.1024. arXiv:2412.16117.

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

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research (TMLR)*, 2025. arXiv:2408.03326.

Duo Li, Zuhao Yang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. ToDRE: Effective visual token pruning via token diversity and task relevance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026. arXiv:2505.18757.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024. arXiv:2311.17005.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, January 2024.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models, 2024. URL `https://arxiv.org/abs/2306.05424`. Also in Proceedings of ACL 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. arXiv:2308.09126.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 932–947, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.66. arXiv:2310.19060.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2403.15388.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2505.21334.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *arXiv preprint arXiv:2507.20198*, 2025b.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2503.11187.

Leqi Shen, Tianxiang Hao, Tao He, Sicheng Zhao, Yifeng Zhang, Pengzhang Liu, Yongjun Bao, and Guiguang Ding. TempMe: Video temporal token merging for efficient text-video retrieval. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025b. arXiv:2409.01156.

Boyuan Sun, Jiaxing Zhao, Xihan Wei, and Qibin Hou. LLaVA-Scissor: Token compression with semantic connected components for video LLMs. *arXiv preprint arXiv:2506.21862*, 2025.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18992–19001, 2025. arXiv:2411.15024.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9961–9980, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.505. Method name: DART (Duplication-Aware Reduction of Tokens). arXiv:2502.11494.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2410.17247.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, 2025. arXiv:2412.04467.

Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 543–553, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-demo.49. arXiv:2306.02858.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2506.10967.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025b. arXiv:2410.04417.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13691–13701, 2025. arXiv:2406.04264.

SUPPLEMENTARY MATERIAL

This supplementary material provides implementation and evaluation details, additional experimental results, and analyses of the scoring space, selection criterion, and temporal redundancy weight.

CONTENTS

A AI Usage Disclosure **14**

B Matched-Budget Protocol across Baselines **14**

C Budget Allocation across Time **14**

D Prompts for the Generative Benchmarks **15**

D.1 Video-ChatGPT 15

D.2 MLVU 16

D.3 Judge configuration 16

E Robustness to Frame Count: Results under 64-Frame Sampling **16**

F Implementation Details of the Scoring Space **17**

G What This Appendix Establishes **18**

H Geometry of the Scoring Space **19**

I Is the Advantage Merely Implicit Centering? **20**

J Separating the Space from the Criterion **21**

K Composition of the Selected Subset **22**

L Temporal Coupling λ_T **23**

M Measurement Hygiene and the MVBench Null Result **24**

N A Falsifiable Prediction: A Backbone Without the Discontinuity **25**

O Extended Coverage **25**

A AI USAGE DISCLOSURE

In this work, we used generative AI tools to assist with polishing the manuscript text. We have carefully reviewed and verified all AI-assisted content and data processing. We take full responsibility for the final content of this work, including all text, claims, analyses, experimental results, and artefacts produced with the assistance of generative AI.

B MATCHED-BUDGET PROTOCOL ACROSS BASELINES

Same budget is not a well-defined constraint across token-compression methods, because different methods parameterise their budget at different points in the network. Our method, together with the sampling baselines, keeps a fixed fraction ρ of the visual tokens once, before the LLM, and never prunes again; its per-layer retention profile is therefore flat at ρ. Baselines that either split their budget across several named quantities, or place a second compression stage inside the LLM, need an explicit conversion. We describe the two cases we had to handle; in both we convert rather than copy the nominal ratio, and we report the visual-token counts actually emitted so that the matching can be audited.

Case 1: a budget split across named quantities. VisionZip splits its budget into *dominant* and *contextual* tokens, both selected before the LLM. The conversion is exact: we set the two quantities to sum to ρ, using $0.0375 + 0.0125$, $0.075 + 0.025$ and $0.15 + 0.05$ for $\rho \in \{5\%, 10\%, 20\%\}$, keeping the upstream 3:1 split between them. The retention profile is flat at ρ, as ours is, so prefill token count and cross-layer average agree and no further calibration is needed.

Case 2: a second compression stage inside the LLM. FlashVID compresses twice: a vision-side stage before the LLM, and an attention-based pruning stage at an intermediate LLM layer. Two adjustments are required. First, its vision-side budget is not ρ: it is computed as $\lceil(\text{tokens per frame}) \times \texttt{retention} \times \texttt{expansion}\rceil$ with a default expansion factor of 1.25, followed by spatio-temporal merging. Passing $\texttt{retention} = \rho$ therefore emits substantially more visual tokens than the nominal ratio suggests — at $\rho = 10\%$ it yields 1445 tokens where ours and VisionZip both yield 1037, a 39% larger budget. We therefore calibrate the parameter empirically, measuring the emitted token count and solving for the value that matches ours: $\texttt{retention} = 0.040/0.079/0.159$, which yields $523/1042/2079$ tokens against our $518/1037/2074$. Second, we set $\texttt{llm_retention} = 1.0$ so that all of its compression happens before the LLM, which makes its retention profile flat and directly comparable to ours.

When we additionally evaluate the in-LLM stage on its own — replacing the vision-side compression by an identity map and pruning only at layer 20 — the retention profile is by construction *not* flat, and prefill token count and cross-layer average no longer agree. That configuration is not budget-matched to ours in the prefill sense, and we report it separately together with its cross-layer compute share (0.73–0.77 of the uncompressed cost) rather than placing it in the same column as the flat-profile methods.

C BUDGET ALLOCATION ACROSS TIME

How the same total budget is distributed over time slices is shown in Figure 4. On the left, the orange curve is STORS with $\lambda_T = 1$, the setting used throughout. The blue curve is per-frame selection, flat because it divides the budget equally by construction. The grey curve keeps the unified pool but removes the temporal term with $\lambda_T = 0$. The green dotted curve, read on the right axis, is the novelty of each slice, taken as the mean over spatial positions of the residual energy a token leaves after being projected onto the token at the same position in the preceding slice. The three allocations differ in the information each acts on. Among the three, $\lambda_T = 1$ follows the novelty curve most closely, since a token explained from another slice is then fully discounted and the budget shifts towards slices carrying content not yet retained. The right panel shows the same tendency at the token level. Per-frame selection returns a similar spatial pattern and the same count in every slice, whereas STORS spends fewer tokens on slices that repeat their predecessor and reallocates them where the scene changes, so the positions it retains across slices cover more complementary regions.

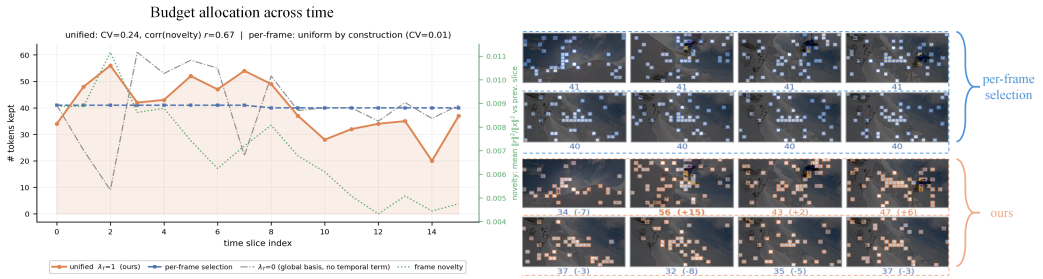


Figure 4: Budget allocation across time under a fixed total budget. Left, tokens kept per time slice for three selection rules, together with the novelty of each slice on the right axis. Right, the positions selected on four slices by per-frame selection (blue) and by STORS (orange), with the signed difference from the per-frame budget annotated under each slice.

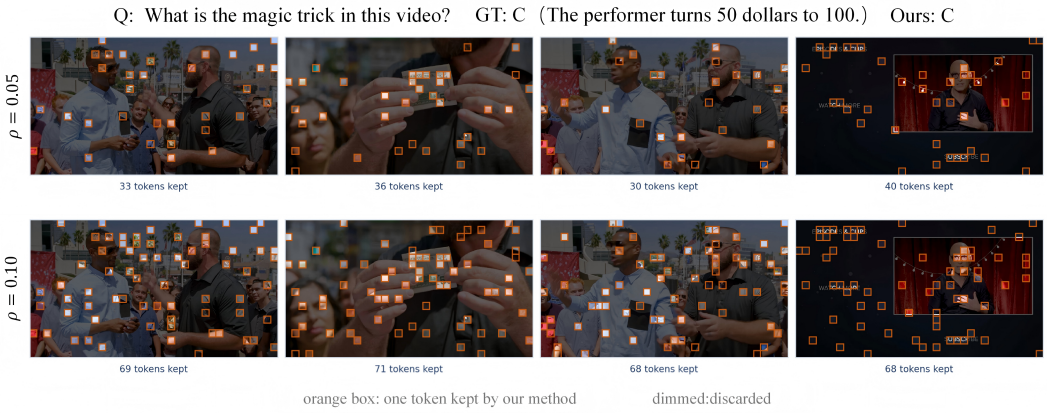


Figure 5: A qualitative example of STORS on Video-MME. Four sampled time slices are shown at two retention ratios.

One example is given in Figure 5. The budget is not split evenly across slices. The close-up of the hands, and the last slice where the scene changes, receive more tokens than the crowd shots. The kept tokens gather on the hands and the cards, which is the evidence the question needs. Selection never sees the question and uses no attention weight. Content of this kind is hard to reconstruct from the rest of the video, so the residual criterion keeps it. Background coverage stays sparse but present, so the surrounding context is not lost. Raising the ratio from 5% to 10% refines the same regions rather than moving the budget elsewhere.

D Prompts for the Generative Benchmarks

All prompts below are reproduced verbatim from our evaluation scripts. Every method in a given table sees the identical prompt, frame sampling and decoding configuration; only the compression module differs. Decoding is greedy throughout.

D.1 Video-ChatGPT

We uniformly sample 32 frames per video. The model is prompted with

```
Carefully watch the video and pay attention to the cause and
sequence of events, the detail and movement of objects, and the
action and pose of persons. Based on your observations, answer
the given question in 1-2 sentences.
```

```
Question: {question}
Answer:
```

with max_new_tokens= 256. Following the benchmark protocol we score five dimensions—*correctness*, *detail*, *context*, *temporal* and *consistency*—with an LLM judge. The consistency dimension is the only one that takes two questions about the same video and both predictions in a single judging call; the other four take one question, the reference answer and one prediction. Each judging call returns a single integer score from 1 to 5, requested in the fixed format

```
Please generate the response in the form of a Python dictionary
string with keys 'score', where its value is an INTEGER from 1
to 5 (5 = best). Only provide the Python dictionary string.
For example: {"score": 4}
```

D.2 MLVU

We uniformly sample 32 frames per video. MLVU mixes seven multiple-choice tasks with two generative ones (8_sub_scene and 9_summary), which use different prompts and different decoding budgets.

Multiple-choice tasks.

```
Carefully watch this video and pay attention to every detail.
Based on your observations, select the best option that
accurately addresses the question.

Question: {question}
Options:
(A) {option A}
...
Only give the best option.
Best option:(
```

Generative tasks.

```
Carefully watch this video and pay attention to every detail.
Based on your observations, answer the following question in
1-3 sentences.

{question}
```

with max_new_tokens= 256. Each generation is scored 1–5 by the same judge, using a task-specific rubric: for 8_sub_scene the judge receives the reference caption together with the annotated scoring points and awards credit per point covered, penalising content that contradicts the reference; for 9_summary it judges accuracy and completeness against the reference summary.

D.3 Judge configuration

Both benchmarks use mimo-v2.5-pro as the judge model, invoked via an OpenAI-compatible endpoint with the temperature set to 0 and the token limit set to 512. We conducted five rounds of evaluation; the values reported in the main text represent the mean, and the variance across the five runs is 0.016.

E Robustness to Frame Count: Results under 64-Frame Sampling

We use Qwen2.5-VL-7B-Instruct and repeat the main experiment at 64 sampled frames, across Video-MME, MLVU, MVBench and EgoSchema. Doubling the frame count doubles the token pool entering the compressor, so at a fixed ρ it also doubles the absolute budget; this is the setting in which the rank ceiling of the score space (Appendix F) binds most often, and it is therefore the sharpest

Table 7: Qwen2.5-VL-7B-Instruct, results under 64-frame sampling.

	Vanilla	Retention Ratio $\rho = 20\%$		Retention Ratio $\rho = 10\%$		Retention Ratio $\rho = 5\%$	
Dataset	(100%)	VisionZip	**Ours**	VisionZip	**Ours**	VisionZip	**Ours**
Video-MME	62.8	**62.1**	61.7	59.8	**61.0**	57.5	**58.4**
MLVU (M-Avg)	64.9	60.9	**62.0**	58.6	**60.9**	56.3	**59.4**
MVBench	67.0	**65.2**	64.3	**62.7**	62.4	**60.1**	60.0
EgoSchema	64.8	61.8	**62.8**	**61.4**	60.2	57.2	**60.6**
Avg. Score	64.9	62.5	**62.7**	60.6	**61.1**	57.8	**59.6**
Rel. Acc. (%)	100.0	96.3	**96.6**	93.4	**94.1**	89.1	**91.8**

available test of whether the reported behaviour is a property of the method or of one particular sampling density.

Table 7 reports overall accuracy on each benchmark. The ordering is preserved at every budget: our method leads VisionZip on average at all three ratios, and the margin behaves exactly as it does at 32 frames, widening as the budget tightens (from $+0.3$ points of relative accuracy at $\rho = 20\%$ to $+2.7$ at $\rho = 5\%$). The same contraction with increasing budget that we document throughout this appendix therefore reappears under a different sampling density, which is what one expects if it is driven by how much redundancy is left for a selection rule to exploit rather than by a particular experimental configuration.

Table 8: Long-video results under 64-frame sampling. Video-MME *long* ($n = 900$) is predominantly 40+ minute clips; MLVU ($n = 2592$, all nine tasks) ranges from 3 minutes to over 2 hours and is reported both on its multiple-choice average (M-Avg, accuracy) and its generative average (G-Avg, judge score out of 5).

		Vanilla	$\rho = 20\%$		$\rho = 10\%$		$\rho = 5\%$	
Benchmark	Metric	(100%)	VisionZip	**Ours**	VisionZip	**Ours**	VisionZip	**Ours**
Video-MME (long)	Acc.	51.4	**50.8**	50.6	49.7	**51.4**	48.8	**49.4**
MLVU	M-Avg	64.89	60.87	**62.03**	58.61	**60.89**	56.25	**59.40**
	G-Avg	2.135	**2.125**	2.093	2.080	**2.116**	2.068	**2.094**

Table 8 isolates the long-video setting. Our method is ahead of VisionZip at $\rho = 5\%$ and $\rho = 10\%$ on both benchmarks. The strongest single result is Video-MME (long) at $\rho = 10\%$, where our method matches the uncompressed upper bound exactly (51.4) using one tenth of the visual tokens. MLVU is the cleanest replication in the whole study: our method leads on the multiple-choice average at all three ratios, retaining 91.5%, 93.8% and 95.6% of the uncompressed score, and the margin again contracts monotonically as the budget grows ($+3.15$, $+2.28$, $+1.16$).

The one cell in which VisionZip is ahead is Video-MME (long) at $\rho = 20\%$, by 0.2 points. We do not attribute this to run-to-run variation: decoding is greedy and the pipeline is deterministic, so the number is exactly reproducible. It is a difference of two questions out of 900, and an exact paired test cannot reject the null hypothesis at that magnitude for any discordant total — the two-sided exact p-value is bounded below by 0.5 whenever $|b - c| = 2$. We therefore describe this cell as a tie rather than as a loss, on the same standard we apply to MVBench in Appendix M.

F Implementation Details of the Scoring Space

Feature construction. STORS performs selection on a token matrix whose columns are the per-token features z_i. We distinguish two spaces. The *reserve space* is the post-projector representation that the LLM actually consumes. The *score space* is the pre-projector representation produced by the vision encoder. Selection, budget accounting and write-back always happen at the LLM-token level, so the two spaces differ only in which features drive the scoring; the visual token budget is identical by construction.

For Qwen2.5-VL the projector is a *merger* that concatenates the four sub-tokens of a 2×2 spatial cell and passes them through an MLP. We therefore obtain the score-space features by a forward pre-

hook on the merger, which captures its input before the MLP is applied, and we average the four sub-tokens within each cell. A concatenating variant that keeps all four sub-tokens (i.e. exactly the MLP input) is used in the mechanism ablation. For LLaVA-OneVision the projector acts per token, and the pre-projector representation is taken directly. Table 9 lists the resulting feature dimensionality for both backbones.

Table 9: Feature dimensionality of each space. d' denotes the dimension actually used for scoring; it equals D unless random projection is enabled (disabled in all reported runs).

Backbone	Space	Definition	$D = d'$
Qwen2.5-VL-7B	reserve	post-merger LLM embedding	3584
	score	pre-merger ViT state, 2×2 cell mean	1280
	score (concat)	pre-merger ViT state, 2×2 concatenated	5120
LLaVA-OneVision-7B	reserve	post-projector LLM embedding	3584
	score	pre-projector encoder state	1152

Temporal slicing and budgets. All main results use 32 sampled frames. Qwen2.5-VL merges two consecutive frames into one temporal patch, so 32 frames correspond to $T = 16$ temporal slices. Table 10 lists the number of visual tokens entering the compressor and the resulting budget $M = \mathrm{round}(\rho TN)$ per segment; all values are read back from the runs themselves rather than computed from the nominal ratio. Because the score space has $d' = 1280$, a budget $M > d'$ cannot be filled by linearly independent columns alone; the excess is necessarily completed by coverage sampling. At 32 frames this ceiling is reached only in the short and long splits of Video-MME at $\rho = 20\%$, and we quantify its effect in Appendix K.

Table 10: Tokens entering the compressor and per-segment budget M at each retention ratio, at 32 sampled frames. Entries exceeding the rank ceiling $d' = 1280$ of the score space are marked †.

Dataset (split)	tokens in	$M_{\rho=5\%}$	$M_{\rho=10\%}$	$M_{\rho=20\%}$
Video-MME (Short / Long)	11520	576	1152	$2304^\dagger$
Video-MME (Medium)	4784	239	478	957
MVBench	2992	150	299	598
EgoSchema	3536	177	354	707

G What This Appendix Establishes

The evidence in this appendix supports a claim with three parts, which we separate deliberately because they have very different effect sizes and very different scopes.

1. **A discontinuity exists at the projector.** The representation the vision encoder produces and the representation the language model consumes have qualitatively different geometry, and the difference is created by the projector itself (Appendix H).
2. **Selecting before the projector is worth a large amount, where that discontinuity is present.** On Video-MME with Qwen2.5-VL, moving the selection from the post-projector to the pre-projector representation is worth $+2.3$ to $+7.6$ points at a fixed budget, positive in all nine cells and significant in eight (Table 11).
3. **The residual criterion is the appropriate rule *for that* geometry, not a better rule in general.** Its contribution is roughly an order of magnitude smaller than the effect of the space, and — more informatively — its sign depends on which space it runs in (Appendix J).

The third point is what makes the first two more than an engineering observation, and it also generates a falsifiable prediction: on a backbone whose projector creates no discontinuity, the choice of space should make no difference at all. Appendix N tests that prediction and confirms it.

We state plainly what Table 11 shows. The effect of the scoring space is large and unambiguous on Video-MME with Qwen2.5-VL, and it is indistinguishable from zero in every other cell we measured: 21 of the 30 cells are null results. We do not regard this as a limitation to be minimised.

Table 11: Effect of moving the selection from the reserve space to the score space, with the criterion, the budget and every hyper-parameter held fixed. Δ is in accuracy points; *, **, *** mark $p < 0.05$, $p < 0.01$, $p < 10^{-3}$ under an exact paired test. The effect is confined to one dataset–backbone combination, which is what the mechanism predicts.

Backbone	Dataset (split)	$\rho = 5\%$	$\rho = 10\%$	$\rho = 20\%$
Qwen2.5-VL-7B	Video-MME (Short)	$\mathbf{+5.56}^{***}$	$\mathbf{+5.67}^{***}$	$\mathbf{+7.56}^{***}$
	Video-MME (Medium)	$+2.33$	$\mathbf{+3.89}^{**}$	$\mathbf{+3.33}^{**}$
	Video-MME (Long)	$\mathbf{+4.22}^{***}$	$\mathbf{+4.33}^{***}$	$\mathbf{+2.67}^{*}$
	MVBench	-0.45	-0.28	-0.55
	EgoSchema (Subset)	$+0.80$	$+0.20$	-1.00
LLaVA-OneVision-7B	Video-MME (Short)	-0.33	$+0.22$	-0.56
	Video-MME (Medium)	$+0.45$	-0.33	$+0.33$
	Video-MME (Long)	-0.78	-0.22	$+0.22$
	MVBench	-0.80	$+0.20$	$+0.10$

A mechanism that predicted a uniform improvement everywhere would be a weaker mechanism, because it would be compatible with almost any explanation. The one we propose says exactly where the effect should appear, and the null cells are where it says it should not.

H Geometry of the Scoring Space

The central empirical claim behind STORS is that the pre-projector space possesses a geometry that differs qualitatively from the post-projector space, and that this difference is created by the projector itself. This appendix characterises that geometry directly, and then rules out the most natural alternative explanation for it.

Anisotropy and cosine concentration. We measure, on held-out videos, the norm structure and the pairwise cosine distribution of the token matrix in each space. Table 12 reports the ratio $\|\mu\|/\mathbb{E}\|x\|$ between the norm of the mean token and the mean token norm, which is a direct scalar summary of anisotropy, together with the cosine statistics before and after removing the mean component.

Table 12: Geometry of the token matrix (Qwen2.5-VL-7B), averaged over held-out videos. The score space is strongly anisotropic and its pairwise cosines are concentrated near one; the reserve space is not. Removing the mean component disperses the score space but does not make it resemble the reserve space.

Space	D	$\mathbb{E}\|x\|$	$\|\mu\|/\mathbb{E}\|x\|$	raw $\cos(x_i, x_j)$		mean-removed cos	
				median	>0.9	median	>0.9
reserve	3584	46.0	0.516	0.265	0.01%	-0.016	0.01%
score	1280	3088.6	**0.998**	**0.997**	**99.98%**	0.284	39.6%
score (concat)	5120	7725.5	0.801	0.685	13.8%	0.023	3.7%

Three observations follow. First, the score space is close to rank one: the mean token accounts for almost the entire token norm, and essentially every token pair has cosine above 0.9. This is precisely the regime in which similarity-based selection criteria lose resolution, because the quantity they rank is nearly constant across candidates. A residual-based criterion is not subject to this failure mode, since it ranks candidates by the component that remains *after* projecting out what has already been selected, and that component is informative even when raw similarities are saturated. The geometry of the score space is therefore not an obstacle for STORS but the condition under which its criterion is the appropriate one.

Second, the concentration does not vanish when the shared mean direction is removed. After mean removal the score space still places 39.6% of pairs above cosine 0.9, against 0.01% in the reserve space — a gap of more than three orders of magnitude. The anisotropy is thus not a thin shell around an otherwise ordinary distribution; structure survives the first-order correction, which is what makes

the residual criterion worth running in this space at all. We return to this point in Appendix I, where we test the same question behaviourally.

Third, the concatenating variant sits between the two spaces on every statistic. It carries the same information and the same spatial granularity as the merger input, and differs from the score space only in that the four sub-tokens are not averaged. Its intermediate geometry is consistent with the mechanism ablation in the main paper, which attributes the discontinuity to the projector MLP rather than to pooling or to spatial merging.

Cross-dataset stability. The measurements in Table 12 were obtained on a different benchmark from the one used for Figure 3 of the main paper, with an independent implementation of the feature extraction, and the two agree to within 0.01: the median pairwise cosine is 0.265 here against 0.275 there in the reserve space, and 0.997 against 0.998 in the score space. The near-rank-one character of the score space likewise reproduces (here $\|\mu\|/\mathbb{E}\|x\| = 0.998$; in the main paper a leading eigenvalue share of 0.999 with an effective rank of 1). The geometry is thus a property of the representation rather than of any particular video distribution.

What this section establishes is only that the two representations are geometrically different. Whether that difference is what produces the accuracy gap, and whether the criterion we pair with it is doing independent work, are separate questions, taken up in Appendices I and J respectively.

I Is the Advantage Merely Implicit Centering?

A natural objection to the analysis above is that the anisotropy of the score space is a trivial artefact of a shared mean component, and that the first step of a greedy pivoted decomposition already removes it. If that were the whole story, explicitly centering the reserve-space features should recover the advantage of the score space. We test this directly by adding a centering option that subtracts the per-dimension mean across tokens before scoring, and applying it to both spaces (Table 13).

Table 13: Centering ablation on Video-MME (Short), $n = 900$ per cell. Explicit centering does not move the reserve space towards the score space; the gap between the two spaces is preserved or slightly widened. Centering the score space is likewise almost free, and never significant. Differences are computed before rounding; p is a two-sided exact paired McNemar test of the centered score space against the uncentered one.

ρ	reserve space			score space				gap: orig. → centered
	default	+center	Δ	default	+center	Δ	p	
5%	60.78	59.78	−1.00	**66.33**	66.11	−0.22	0.868	+5.56 → +6.55
10%	63.67	63.00	−0.67	**69.33**	69.00	−0.33	0.780	+5.67 → +6.33
20%	63.67	63.78	+0.11	**71.22**	70.56	−0.67	0.471	+7.56 → +7.44

The outcome is unambiguous in both directions, and in the direction that favours the mechanism we propose. Centering the reserve space does not close the gap: it *widens* it at the two tighter budgets and leaves it essentially unchanged at the loosest, so the last column moves from $+5.56/+5.67/+7.56$ to $+6.55/+6.33/+7.44$. Centering the score space does not destroy its advantage either; the cost is a few tenths of a point and is not significant at any budget. The shared mean component is therefore neither the source of the effect nor the carrier of it.

That centering is not a no-op is worth emphasising, because it is what makes the null result informative. The operation changes the geometry of the score space substantially: it increases the mean relative residual by roughly an order of magnitude (from 5×10^{-4} to 4.9×10^{-3}) and drives the fraction of budget obtained by greedy selection from 0.92 to 1.00, yet accuracy moves by less than a point. The structure that STORS exploits therefore lives in the directions that survive mean removal — consistent with the mean-removed cosine statistics of Table 12, and with a projector-MLP mechanism rather than a first-order statistical artefact.

J SEPARATING THE SPACE FROM THE CRITERION

Appendix G reports the effect of the scoring space with the criterion held fixed. This section does the complement: it holds the space fixed and varies the criterion, in *both* spaces. The comparison baselines are a uniform spatio-temporal grid and a random subset at the same budget, run with identical hyper-parameters to the main configuration so that only the selection rule differs. All arms of a comparison run on the same host against the same copy of the data and are tested pairwise.

Running this in both spaces is what makes the experiment informative. If the residual criterion were simply a better selection rule, it would beat a uniform grid wherever it ran. Table 14 shows that it does not.

Table 14: The same residual criterion, run in each of the two spaces, against the same budget-matched baselines (Qwen2.5-VL-7B). Δ is measured against the uniform grid; *, **, *** mark $p < 0.05$, $p < 0.01$, $p < 10^{-3}$ under an exact paired test. On Video-MME the criterion is *worse* than a uniform grid in the reserve space and better in the score space; on the two benchmarks with no space effect it is mildly positive in both.

Dataset (split)	ρ	baselines		residual criterion		Δ vs. grid	
		grid	random	in reserve	in score	in reserve	in score
Video-MME (Short)	5%	63.67	63.67	60.78	**66.33**	-2.89^{*}	$+2.67^{*}$
	10%	67.33	66.11	63.67	**69.33**	-3.67^{**}	$+2.00$
	20%	69.78	70.22	63.67	**71.22**	-6.11^{***}	$+1.44$
Video-MME (Medium)	5%	52.33	52.89	50.22	**52.56**	-2.11	$+0.22$
	10%	55.22	54.89	51.56	**55.44**	-3.67^{**}	$+0.22$
	20%	55.89	56.22	53.00	**56.33**	-2.89^{*}	$+0.44$
Video-MME (Long)	5%	47.00	47.00	45.78	**50.00**	-1.22	$+3.00^{**}$
	10%	49.22	49.56	47.22	**51.56**	-2.00	$+2.33^{*}$
	20%	50.78	51.33	48.78	**51.44**	-2.00	$+0.67$
MVBench	5%	57.50	57.28	**59.03**	58.58	$+1.53^{**}$	$+1.08^{*}$
	10%	60.38	59.91	**61.53**	61.26	$+1.15^{*}$	$+0.88$
	20%	63.06	62.38	**63.68**	63.13	$+0.63$	$+0.08$
EgoSchema (Subset)	5%	55.80	54.80	55.40	**56.20**	-0.40	$+0.40$
	10%	59.40	57.80	60.00	**60.20**	$+0.60$	$+0.80$
	20%	61.20	60.80	**62.00**	61.00	$+0.80$	-0.20

The table shows that the measured performance of residual selection depends on the scoring space. On Video-MME the residual criterion is negative in all nine cells when it runs in the reserve space — significantly so in five, and by as much as 6.1 points — and positive in all nine when the identical criterion runs in the score space. Nothing changes between the two columns except which representation the scores are computed from. A rule that is worse than a uniform grid in one representation and better in the other is not usefully described as a better selection rule; it is a rule whose validity is a property of the representation it is applied to. This is why we present the space as the finding and the criterion as the instantiation, rather than the other way round.

On MVBench and EgoSchema, where Table 11 shows no space effect, there is correspondingly no interaction: the criterion is mildly positive in both spaces (MVBench $+0.63$ to $+1.53$ in the reserve space, $+0.08$ to $+1.08$ in the score space). This is the expected pattern if the two representations are not meaningfully different there, and it also shows that the criterion is not inherently harmful in the reserve space — only where the discontinuity is large.

Two further regularities are worth recording, both consistent with the account above. First, within the score space the advantage over the baselines contracts monotonically with the budget on four of the five benchmarks, and significance contracts with it: six of ten comparisons reach $p < 0.05$ at $\rho = 5\%$, four at $\rho = 10\%$, none at $\rho = 20\%$. When a fifth of the tokens are kept, almost any subset spans most of the useful signal and no selection rule can distinguish itself; the regime in which selection is worth doing at all is the aggressive one. Second, the margin over the grid within the score space is smallest on the Medium split ($+0.22$ to $+0.44$, never significant), which enters the compressor with less than half the token pool of the other two Video-MME splits. We report

this rather than restricting the comparison to favourable splits. It bears on the size of the criterion's contribution, not on the space effect, which on that same split is $+2.33$ to $+3.89$ and significant at two of three budgets.

K Composition of the Selected Subset

STORS completes its budget with coverage sampling whenever the greedy stage becomes rank-deficient, and switches over early when the relative residual of the next candidate falls below a fixed threshold. Since a uniform grid is itself a baseline (Appendix J), it is important to state how much of the budget each stage contributes, and to establish whether the coverage stage is responsible for the reported gains. We report both. The question matters for the space effect as much as for the criterion: the greedy fraction differs systematically between the two spaces, so a reviewer is entitled to ask whether the space effect is really a difference in how the budget gets filled.

Table 15: Fraction of the budget obtained by greedy selection under the default configuration. The same fixed threshold produces full greedy selection in the reserve space and early switching in the score space, because the relative residual is two to three orders of magnitude smaller there.

Dataset (split)	Space	$\rho = 5\%$	$\rho = 10\%$	$\rho = 20\%$
Video-MME (Short)	score	0.92	0.50	0.33
Video-MME (Medium)	score	1.00	0.75	0.33
Video-MME (Long)	score	0.67	0.33	0.25
MVBench	score	0.35	0.25	0.25
EgoSchema	score	0.51	0.25	0.17
all of the above	reserve	1.00	1.00	1.00

The mechanism behind Table 15 is the geometry of Appendix H, not a tuned choice. Against the same fixed threshold of 0.05, the mean relative residual is 5–9 $\times\, 10^{-4}$ in the score space and 1.0×10^{-2} to 3.2×10^{-1} in the reserve space, so the switch fires almost immediately in one and never in the other. A large grid share in the score space is thus a *symptom* of the near-rank-one geometry we rely on, and the natural question is whether it is also the cause of the accuracy.

It is not. Disabling the early switch removes the threshold entirely and yields a parameter-free variant that runs greedy selection as far as the rank of the token matrix allows. Under that variant the greedy fraction obeys $\min(1,\, d'/M)$ exactly: it is 1.00 wherever $M \le d' = 1280$, and equals the rank ceiling otherwise — for instance 0.555 measured against a predicted 0.556 at $M = 2304$, and 0.278 against 0.278 at $M = 4608$ under 64-frame sampling. The composition of the subset is therefore fully determined and can be moved from roughly a quarter to one at will. Table 16 reports what that movement costs.

Table 16: Accuracy with and without the early switch to coverage sampling, 9 matched pairs. Moving the greedy fraction from as low as 0.25 up to 1.00 changes accuracy by at most 1.22 points, in inconsistent directions, and no comparison is significant.

Dataset (split)	ρ	default	full greedy	Δ	b/c	p
MVBench	5%	58.58	59.13	$+0.55$	142/164	0.230
MVBench	10%	61.26	61.41	$+0.15$	148/154	0.774
MVBench	20%	63.13	63.53	$+0.40$	131/147	0.368
Video-MME (Short)	5%	66.33	66.56	$+0.22$	10/12	0.832
Video-MME (Short)	10%	69.33	68.11	-1.22	36/25	0.200
Video-MME (Short)	20%	71.22	71.67	$+0.44$	23/27	0.672
Video-MME (Medium)	10%	55.44	54.78	-0.67	33/27	0.519
Video-MME (Long)	5%	50.00	50.00	0.00	10/10	1.000
Video-MME (Long)	10%	51.56	51.22	-0.33	22/19	0.755

Across all nine matched pairs the accuracy difference never exceeds 1.22 points, does not have a consistent sign, and is not significant in a single cell, while the greedy fraction changes by a factor of up to four. Whether a token enters the subset because it maximises the orthogonal residual or because it completes the coverage grid is thus largely immaterial at a fixed budget: the two stages are interchangeable rather than one carrying the other, and neither the advantage over the baselines

in Appendix J nor the gap between the two scoring spaces can be attributed to the coverage stage. In particular, the reserve space runs at a greedy fraction of 1.00 everywhere and the score space at 0.17–1.00, yet the accuracy difference between them reaches $+7.6$ points; since moving the greedy fraction across that same range at a fixed scoring rule is worth nothing, the difference between the spaces is not a difference in subset composition.

We emphasise that we are not arguing for the early switch, only reporting that it is inert. Because the two settings are statistically indistinguishable in every cell we tested, none of the conclusions in this paper depends on the threshold, and a reader who prefers the parameter-free variant may read the "full greedy" column throughout without any claim changing. We retain the switch as the default because it is cheaper — it stops the pivoted decomposition once the residual is numerically negligible — and not because it is required for accuracy.

L Temporal Coupling λ_T

Full sweep. Table 17 extends the sweep of the temporal coupling beyond the reported operating point. The response is unimodal on all three splits: a flat plateau below one, a rise at $\lambda_T = 1$, and a decline above it that falls below the plateau. The shape is identical across the three splits, which is what one expects of a parameter that is doing something structural rather than fitting noise.

Table 17: Accuracy as a function of the temporal coupling (score space, $\rho = 10\%$, $n = 900$ per cell). The optimum at $\lambda_T = 1$ is a genuine interior maximum, not the endpoint of a monotone trend. The last row gives the fraction of the budget obtained by greedy selection on the Short split.

Split	λ_T=0	0.25	0.5	0.75	**1.0**	1.25	1.5	2.0
Short	66.00	66.00	66.11	66.33	**69.33**	64.22	63.78	64.44
Medium	54.11	54.44	54.00	54.22	**55.44**	52.11	52.00	52.44
Long	49.78	49.67	49.67	49.67	**51.56**	47.67	47.78	46.78
greedy frac. (Short)	1.00	1.00	1.00	1.00	0.50	0.67	0.67	0.67

Controlling for selection composition. Because $\lambda_T = 1$ also alters how quickly the greedy stage saturates, we verify that the optimum is not an artefact of a change in the greedy fraction. Two independent facts rule this out. First, the ordering of accuracies does not follow the ordering of greedy fractions: the settings above one have an *intermediate* greedy fraction (0.67–0.75 across splits) and the *worst* accuracy, so no monotone relation between the two can explain the peak. Second, comparing $\lambda_T = 0$ and $\lambda_T = 1$ with the greedy fraction held at one on both sides preserves the advantage on all three splits (Table 18), although with a smaller margin than the unmatched comparison. We report the matched figures as the honest estimate of the effect of the temporal term.

Table 18: Effect of the temporal coupling with and without holding the selection composition fixed ($\rho = 10\%$, $n = 900$ per cell). In the matched comparison the greedy fraction equals 1 on both sides, so the two arms differ only in the temporal term. p-values are two-sided exact paired tests.

Split	unmatched			matched (greedy frac. = 1 both sides)		
	λ_T=0	λ_T=1	Δ (p)	λ_T=0	λ_T=1	Δ (p)
Short	66.00	**69.33**	+3.33 (0.006)	66.00	**68.11**	+2.11 (0.064)
Medium	54.11	**55.44**	+1.33 (0.276)	54.11	**54.78**	+0.67 (0.634)
Long	49.78	**51.56**	+1.78 (0.060)	49.78	**51.22**	+1.44 (0.160)

Per-task behaviour. Aggregate optima can hide task-level heterogeneity, so we decompose the sweep by task on a benchmark with explicit task labels (Table 19). In three of the six tasks a coupling of one is optimal or tied for optimal. The two clear exceptions are the tasks that turn on egocentric orientation and movement direction, which strictly prefer no temporal coupling and account for the aggregate preference for $\lambda_T = 0$ on this benchmark. This is interpretable, and it is the same distinction that organises Appendix J: those tasks are decided by the relative ordering of a few frames rather than by diverse coverage across time, so spreading the budget temporally buys nothing and costs resolution where it is needed. The reading we take from this is that λ_T is a dataset-dependent knob rather than a universal constant, which we state as a limitation; the value $\lambda_T = 1$

used throughout is selected on Video-MME and held fixed everywhere else rather than tuned per benchmark.

Table 19: Per-task accuracy across the temporal coupling ($\rho = 20\%$, $n = 200$ per task). Tied optima are both listed in the last column.

Task	$\lambda_T = 0$	$\lambda_T = 1$	$\lambda_T = 1.5$	best
Action Prediction	**67.00**	65.50	**67.00**	0 / 1.5
Action Sequence	74.50	**76.00**	75.50	1.0
Counterfactual Inference	**62.00**	**62.00**	59.50	0 / 1.0
Egocentric Navigation	**38.50**	36.50	34.00	0
Episodic Reasoning	51.50	**52.00**	**52.00**	1.0 / 1.5
Moving Direction	**47.50**	45.50	43.00	0
aggregate	**56.83**	56.25	55.17	0

M MEASUREMENT HYGIENE AND THE MVBENCH NULL RESULT

The null cells in Table 11 carry weight in our argument, so they have to be null for the right reason. This section reports the one case where we found that an apparent effect was an artefact of measurement rather than of the method, and states the standard we apply to all of the comparisons in this paper.

All comparisons reported in the main paper are run on a single host with an identical copy of each dataset. We found that this is not a formality: an earlier version of our MVBench comparison mixed two hosts whose copies of the benchmark differ in the encoding of one subset, and the resulting differences were inflated by a factor of two to four in both of the comparisons we were able to check. Table 20 reports the same-host reproduction together with exact paired McNemar tests, with items matched by (task, video, question).

Table 20: MVBench, same host, all 20 tasks. Every run answers all 4000 items; two (task, video, question) triples occur twice in the benchmark itself, so pairing is over the $n = 3998$ distinct keys. Neither the choice of scoring space nor the comparison against the strongest baseline reaches significance at any budget: on this benchmark the methods are statistically tied. b counts items the first system alone gets right, c items the second system alone gets right.

Comparison	ρ	accuracy	Δ	b/c	p
reserve vs. score	5%	59.03 / 58.58	-0.45	217/199	0.405
	10%	61.53 / 61.26	-0.28	198/187	0.610
	20%	63.68 / 63.13	-0.55	194/172	0.272
VisionZip vs. STORS	5%	58.73 / 58.58	-0.15	277/271	0.831
	10%	61.83 / 61.26	-0.58	220/197	0.281
	20%	63.76 / 63.13	-0.63	189/164	0.201

Two consequences follow, and both matter for the claim structure of Appendix G. First, MVBench is a clean null for the scoring space rather than a negative result: an earlier version of this work reported the space as significantly harmful here, and that finding did not survive same-host measurement. The dataset–backbone table therefore contains large positive cells and null cells, with no negative cell anywhere — which is what a mechanism that predicts "effect where the discontinuity exists, no effect otherwise" requires. A genuinely negative cell would have falsified it.

Second, we apply the same standard everywhere: claims of the form "ahead of the strongest baseline" are restricted to comparisons that pass a paired test. On MVBench we describe the comparison against the strongest baseline as a tie rather than a small loss, and we apply the identical standard to the one cell in which VisionZip is nominally ahead at 64 frames (Appendix E).

N A Falsifiable Prediction: A Backbone Without the Discontinuity

The account in Appendix H attributes the discontinuity to the projector, and specifically to the fact that Qwen2.5-VL's projector is a *merger*: it concatenates the four sub-tokens of a spatial cell and passes them jointly through an MLP. That construction is not universal. LLaVA-OneVision projects each token independently, so its pre- and post-projector representations are related by a per-token map that leaves the spectrum and the pairwise cosine distribution essentially unchanged.

This yields a prediction that can fail. If the effect we report were a generic benefit of scoring earlier in the network — lower dimension, less task-specialisation, whatever the reader prefers — it should reappear on LLaVA-OneVision. If it is specifically the projector-induced discontinuity, it should vanish there entirely, and the method should still work, just with nothing to gain from the change of space.

Table 21 runs the full grid: three duration splits, three budgets, paired tests throughout. The prediction holds. The scoring space has no effect at any operating point: nine cells, all $|\Delta| \leq 0.78$, all $p \geq 0.45$, with signs distributed evenly in both directions. Together with the MVBench null in Table 11, this means the space effect is confined to the cells in which the mechanism says a discontinuity exists, and is absent — not merely small, and never negative — everywhere else.

It is worth being explicit that this is the outcome we would have had to report either way, and that a positive result here would have been bad news for the explanation while being good news for the headline number. The lower half of the table also shows that the method itself transfers to this backbone without difficulty, retaining 98.8% of the uncompressed score at $\rho = 20\%$ and 92.2% at $\rho = 5\%$; what does not transfer is the benefit of changing the space, which is exactly the claim.

Table 21: LLaVA-OneVision-7B on Video-MME. Top: the two scoring spaces are indistinguishable at every operating point. Bottom: retention of the uncompressed upper bound under the standard configuration (score space). A handful of items hit transient video-decoder failures; these are excluded from both arms of the affected comparison, which is why some cells have $n < 900$. The bottom block uses one common item set per split, shared by all four configurations.

Split	ρ	reserve	score	Δ	b/c	p	n
Short	5%	66.00	65.67	−0.33	36/33	0.810	900
	10%	66.89	67.11	+0.22	28/30	0.896	900
	20%	71.11	70.56	−0.56	37/32	0.630	900
Medium	5%	51.62	52.06	+0.45	41/45	0.747	897
	10%	54.62	54.28	−0.33	41/38	0.822	899
	20%	55.28	55.62	+0.33	39/42	0.824	899
Long	5%	47.27	46.49	−0.78	35/28	0.450	897
	10%	48.11	47.89	−0.22	34/32	0.902	900
	20%	49.56	49.78	+0.22	29/31	0.897	900

Configuration	Short	Medium	Long	Avg.	retention
Vanilla (100%)	72.11	57.17	48.72	59.33	—
STORS $\rho = 5\%$	65.67	51.95	46.49	54.70	92.2%
STORS $\rho = 10\%$	67.11	54.17	47.94	56.41	95.1%
STORS $\rho = 20\%$	70.56	55.51	49.83	58.63	**98.8%**

O Extended Coverage

Normalisation of the scoring features. Appendix I shows that removing the mean direction from the scoring features is harmless. The complementary operation — unit-normalising each token, which removes magnitude while leaving the mean direction intact — behaves in the opposite way (Table 22). Over the full 5×3 grid it degrades accuracy in 13 of 15 cells, significantly so in 9, and it never produces a significant improvement: the two cells that are not negative are $+0.44$ and $+0.03$ points, with $p = 0.78$ and $p = 1.00$. Token norm therefore carries usable signal in the score space

and must not be normalised away, whereas the shared mean direction carries none. That the two first-order normalisations give opposite answers is itself informative: it rules out the reading that the score space works for some generic reason, and localises what the criterion is actually using.

The mechanism is visible in the selection statistics. Unit-normalisation causes the relative residual to collapse almost immediately, so the greedy fraction falls sharply — for example from 1.00 to 0.25 on the Medium split at $\rho = 5\%$, and from 0.67 to 0.17 on the Long split at the same budget. This is what one expects if magnitude is part of what the criterion ranks: removing it leaves the remaining directions nearly interchangeable. It is not, however, the source of the accuracy loss. Appendix K shows that moving the greedy fraction over the same range at a fixed scoring rule leaves accuracy unchanged, and the two cells here with the lowest greedy fractions have opposite signs ($+0.44$ on Medium at $\rho = 5\%$, -2.89 on Long at $\rho = 5\%$). The harm comes from what the criterion can no longer distinguish, not from how the budget is finally filled.

Four of these cells were additionally re-run from scratch on a second host. The replication agrees in sign and significance in all four, and to within 0.44 points in magnitude ($+0.44$ vs. $+0.78$, -1.33 vs. -1.22, -2.89 vs. -3.33, -2.00 vs. -1.89).

Table 22: Effect of unit-normalising the scoring features ($x \mapsto x/\|x\|$) relative to the default score space, over the full 5×3 grid. Entries in bold are significant at the 0.05 level under an exact paired test. Video-MME cells use $n = 900$ except Short $\rho = 20\%$, where 14 items hit a transient decoder failure and are excluded from both arms ($n = 886$); MVBench uses $n = 3998$ and EgoSchema $n = 500$.

Dataset (split)	$\rho = 5\%$		$\rho = 10\%$		$\rho = 20\%$	
	Δ	p	Δ	p	Δ	p
Video-MME (Short)	$\mathbf{-2.56}$	0.049	$\mathbf{-3.00}$	0.009	$\mathbf{-1.58}$	0.049
Video-MME (Medium)	$+0.44$	0.775	$\mathbf{-2.33}$	0.024	-1.33	0.175
Video-MME (Long)	$\mathbf{-2.89}$	0.007	$\mathbf{-2.56}$	0.010	$\mathbf{-2.00}$	0.015
MVBench	$\mathbf{-1.00}$	0.039	$\mathbf{-1.33}$	0.003	$+0.03$	1.000
EgoSchema (Subset)	-1.00	0.442	-2.00	0.076	-1.40	0.144