COMPLETE THE SCENE, COVER THE VIDEO: RECURRENT TOKEN COMPRESSION FOR VIDEO LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token compression aims to reduce inference costs in Video Large Language Models while preserving information for video understanding. Existing compressors often build on independent frame-wise selection of important tokens. However, such selection may repeatedly retain content already well represented in earlier outputs while overlooking information not yet preserved, limiting coverage of content unfolding throughout the video. Motivated by representation gaps revealed in our video-level feature coverage analysis, we propose C^2V, a training-free, before-LLM method using the preceding frame's compressed representation as a visual state. It greedily selects anchors that complement the state in representing the current scene. Candidate sets are evaluated by Maximum Mean Discrepancy (MMD) between the attention-weighted scene distribution and an optimal convex mixture of the state and candidate anchor distributions. Mass-preserving aggregation incorporates unselected tokens' features and probability mass into the anchors, producing LLM input features and the anchor distribution used as the next visual state. We evaluate C^2V on five benchmarks spanning short clips to long videos, using Qwen3-VL-8B, InternVideo3-8B, and Qwen2.5-VL-7B at four retention ratios (10–25%). C^2V achieves the highest five-benchmark average accuracy among evaluated compressors in all 12 model–retention settings. It also outperforms competing methods in video-level feature coverage at matched token budgets, indicating better collective representation of source features.

1 INTRODUCTION

Video Large Language Models (VLLMs) pass visual tokens from multiple frames to a language model (Lin et al., 2024; Li et al., 2025; Wang et al., 2024; Yan et al., 2026), making longer inputs computationally and memory intensive (Qian et al., 2024; Chen et al., 2025; Weng et al., 2024). Token compression reduces these costs while seeking to preserve understanding. Image methods select or merge tokens using vision-encoder attention or feature similarity (Liang et al., 2022; Fayyaz et al., 2022; Wei et al., 2023; Bolya et al., 2023). These principles extend to independent frame-wise compression (Yang et al., 2025; Shang et al., 2025), with recent video methods further reducing temporal redundancy through inter-frame merging (Shen et al., 2025; Fan et al., 2026).

However, selecting each frame's important tokens independently of other frames' compressed outputs does not necessarily yield a faithful representation of the video as a whole. In videos, content persists across frames as changes and new information emerge. Such selection may repeatedly retain important content already well represented, leaving less of the token budget for information not yet adequately preserved. To examine the resulting collective representation, we analyze the video-level feature coverage of the final compressed outputs from three methods built on independent frame-wise representative selection (Yang et al., 2025; Shen et al., 2025; Fan et al., 2026). We consider unweighted, attention-weighted, and uniqueness-weighted evaluations, which respectively treat source tokens equally, emphasize high-attention tokens, and emphasize tokens distinct in feature space. Figure 1(a–c) shows that C^2V achieves higher coverage than these methods at matched token budgets under all three evaluations, indicating room to improve their collective representation. This motivates selecting current tokens to complement previously retained representations, rather than selecting frame-wise representatives independently. We therefore ask: *Given the preceding compressed representation, what additional information should be retained to adequately represent the current scene?*

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

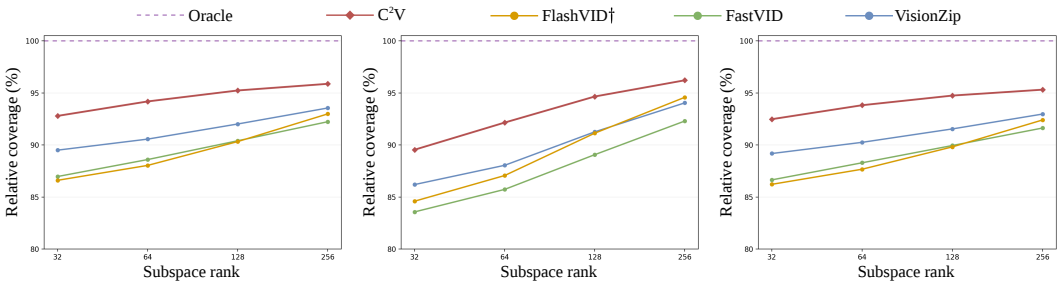

Figure 1: Video-level feature coverage on VideoMME (Qwen3-VL-8B; 32 sampled frames; 10% retention): source-feature reconstruction by rank-r output subspaces under (a) unweighted, (b) attention-weighted, and (c) uniqueness-weighted evaluation. Dashed lines denote optimal rank-r references (100%). See Sec. 4.4.

We propose **C^{2}V** (**C**omplete the Scene, **C**over the **V**ideo), a training-free, before-LLM compressor that uses the preceding frame's compressed representation as a visual state. It treats scene completion as distribution approximation by the preceding state and current anchors. Figure 2 summarizes three stages. **(a) Scene Distribution Construction** represents the current scene using normalized feature directions as support points and normalized attention weights as probability masses. **(b) State-Conditioned Scene Completion** greedily selects anchors by minimizing Maximum Mean Discrepancy (MMD) between the scene distribution and an optimal state–anchor mixture. **(c) Visual State Construction** performs mass-preserving aggregation, incorporating unselected tokens' features into their assigned anchors and transferring their attention mass. The resulting distribution becomes the next visual state for selection, while the LLM receives compressed features from all frames.

We evaluate C^{2}V on MVBench, TempCompass, VideoMME, LongVideoBench, and LVBench using Qwen3-VL-8B, InternVideo3-8B, and Qwen2.5-VL-7B at four retention ratios (10–25%). Results show consistent gains in five-benchmark average accuracy across these settings and higher video-level feature coverage in matched-budget analysis. We also evaluate frame scaling under a fixed LLM visual-token budget and integration with inner-LLM pruning.

Our main contributions are summarized as follows.

- We analyze the video-level feature coverage of compressors built on frame-wise representative selection under three source-token weightings, identifying representation gaps that can be reduced at matched token budgets.
- We formulate C^{2}V as state-conditioned scene completion through MMD anchor selection and mass-preserving aggregation, using a single method-specific hyperparameter fixed across models, benchmarks, and retention ratios.
- C^{2}V achieves the highest five-benchmark average accuracy among the compared methods in all 12 model–retention settings and shows higher video-level feature coverage in our matched-budget analysis.

2 Related Work

2.1 Visual Token Compression

Training-free token compression operates before the LLM, within its layers, or at both stages (hybrid). DivPrune (Alvar et al., 2025) selects diverse visual tokens before the LLM, while FastV (Chen et al., 2024) prunes tokens using early-layer LLM attention. Hybrid methods such as HoliTom and FlashVID (Shao et al., 2025; Fan et al., 2026) combine before-LLM spatiotemporal compression with inner-LLM token reduction. We focus on before-LLM compression.

VisionZip (Yang et al., 2025) applies attention-based selection and similarity-based aggregation within each frame. FastVID (Shen et al., 2025) follows attention-based frame-wise selection with density-based merging of the remaining tokens within temporal segments. FlashVID's before-LLM

stage (Fan et al., 2026) selects representatives using attention, diversity, and event relevance, then applies tree-based inter-frame merging to the remaining tokens. These methods differ in aggregation scope but do not condition representative selection on earlier compressed outputs.

Other methods incorporate past information into selection. MMG-Vid (Ma et al., 2026) allocates segment budgets by marginal gain and selects tokens using dissimilarity to prior selections and within-frame diversity. EchoPrune (Li et al., 2026) combines query relevance with reconstruction error from the preceding frame. C^{2}V instead scores anchor sets by residual MMD after optimally mixing current anchors with the preceding compressed state. Each candidate is evaluated jointly with selected anchors against the full current-scene distribution, so novelty alone does not determine its value.

2.2 Kernel-Based Distribution Approximation

MMD compares distributions through their kernel mean embeddings (Gretton et al., 2012). Kernel herding (Chen et al., 2010) sequentially reduces target-mean approximation error, accounting for samples already selected for the same target. Kernel thinning (Dwivedi & Mackey, 2021) constructs smaller representative subsets while controlling kernel approximation error. C^{2}V applies this set-level approximation principle recurrently, conditioning each current scene's approximation on the preceding frame's compressed state.

3 Method

3.1 Overview

C^{2}V processes frames chronologically, conditioning current anchor selection on the preceding frame's compressed representation. Here, a frame denotes a native temporal unit of the compressor input, with visual features $V_t = \{v_{t,i}\}_{i=1}^{N}$ for $t = 1, \ldots, T$. We distribute the total output token budget nearly uniformly across frames, retaining K_t tokens per frame. At each step, we construct the current scene distribution, select state-conditioned anchors, and aggregate the current tokens to produce compressed LLM input features and the next visual state (Figure 2; Algorithm A.1).

3.2 Scene Distribution Construction

Normalized comparison space. Following cosine-based feature comparison in prior compression methods (Alvar et al., 2025; Yuan et al., 2026), we normalize each nonzero feature as $\hat{x}_{t,i} = v_{t,i}/\rho_{t,i}$, retaining its original norm $\rho_{t,i} = \|v_{t,i}\|_2$ for output scaling in Sec. 3.4. For unit directions $\hat{x}, \hat{y}$, $\|\hat{x} - \hat{y}\|_2^2 = 2(1 - \hat{x}^\top \hat{y})$; hence, the Laplacian kernel in Sec. 3.3 depends on cosine dissimilarity rather than feature magnitude.

Attention-based probability masses. Following attention-based token scoring (Liang et al., 2022; Yang et al., 2025), we use the final vision-encoder self-attention layer. When a [CLS] token is available, we use [CLS]-to-patch attention; otherwise, we aggregate the attention each patch receives from queries within the same frame. After averaging over heads and aligning the scores with the backbone's pooling or spatial merger, we obtain $\alpha_{t,i}$ and normalize it within each frame:

$$a_{t,i} = \frac{\alpha_{t,i}}{\sum_j \alpha_{t,j}}, \qquad P_t = \sum_{i=1}^{N} a_{t,i} \delta_{\hat{x}_{t,i}}. \tag{1}$$

Here, δ_x denotes a unit point mass at x. The scene distribution P_t weights normalized feature directions by attention, so high-attention tokens contribute more to its mean embedding.

3.3 State-Conditioned Scene Completion

MMD-based distribution comparison. Let $\mu_P = \mathbb{E}_{x \sim P}[\kappa(x, \cdot)]$ denote the kernel mean embedding of P. MMD compares distributions in the reproducing kernel Hilbert space $\mathcal{H}_\kappa$ (Gretton et al., 2012):

$$\mathrm{MMD}_\kappa^2(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}_\kappa}^2. \tag{2}$$

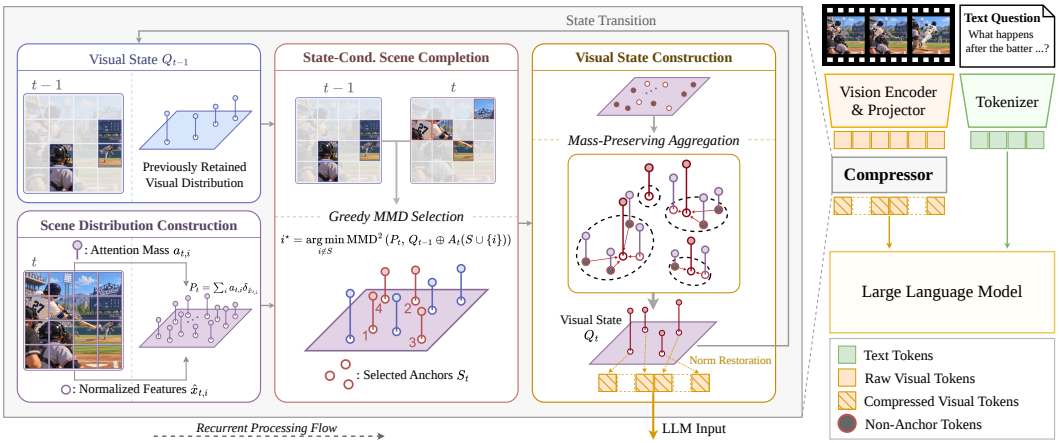

Figure 2: Overview of C^{2}V. Greedy MMD selection identifies anchors that complement the preceding state Q_{t-1} in representing the current scene distribution P_t; $\oplus$ denotes the optimal convex state–anchor mixture. Mass-preserving aggregation incorporates unselected tokens' features and transfers their attention mass to the anchors, producing the next state Q_t and compressed LLM input features.

This compares the dense scene, compressed state, and candidate anchors in one space despite their different supports and probability masses. With a linear kernel, MMD compares only weighted feature means, which can coincide for distinct token distributions. We use a characteristic Laplacian kernel to distinguish such distributions through their kernel mean embeddings (Sriperumbudur et al., 2010):

$$\kappa_\delta(\hat{x}, \hat{y}) = \exp\left[-(\ln 2)\sqrt{\frac{1-\hat{x}^\top \hat{y}}{\delta}}\right] = 2^{-\sqrt{(1-\hat{x}^\top \hat{y})/\delta}}, \qquad \delta > 0. \tag{3}$$

This parameterization yields $\kappa_\delta(\hat{x}, \hat{y}) = 1/2$ when the cosine dissimilarity $1 - \hat{x}^\top \hat{y}$ equals δ. The kernel scale δ is the only method-specific hyperparameter and is fixed across all evaluated models, benchmarks, and retention ratios.

State-conditioned candidate evaluation. We use Q_{t-1}, formed from the preceding frame's aggregated feature directions and attention masses (Sec. 3.4), as the visual state. For any nonempty $S \subseteq \{1, \ldots, N\}$, let $A_t(S) = |S|^{-1} \sum_{s \in S} \delta_{\hat{x}_{t,s}}$ be the uniform distribution over pre-aggregation anchors. Attention weighting enters through P_t; uniform $A_t(S)$ scores anchor locations before aggregation. This surrogate avoids recomputing full-token assignments for every trial set, whereas Sec. 3.4 determines final cluster masses and fused features. We evaluate each set by the residual squared MMD after optimizing its convex state–anchor mixture (Appendix A.3):

$$\mathcal{L}_t(S) = \min_{\lambda \in [0,1]} \mathrm{MMD}^2_\kappa\left(P_t,\, \lambda Q_{t-1} + (1-\lambda) A_t(S)\right). \tag{4}$$

Unlike candidate–state dissimilarity, Eq. (4) measures the residual error in jointly representing the current scene. Since state-only error is independent of S, minimizing $\mathcal{L}_t(S)$ maximizes the candidate set's error reduction. State use is not forced: because $\lambda = 0$ is feasible, the optimum is no worse under this objective than evaluating the same candidates alone and can ignore an unhelpful state.

Since $\mu_{\lambda Q + (1-\lambda)A} = \lambda \mu_Q + (1-\lambda)\mu_A$, Eq. (4) is quadratic in λ. Projecting the current scene embedding μ_{P_t} onto the line segment connecting $\mu_{Q_{t-1}}$ and $\mu_{A_t(S)}$ gives the optimal mixture coefficient:

$$\lambda_t^\star(S) = \left[\frac{\left\langle \mu_{P_t} - \mu_{A_t(S)},\, \mu_{Q_{t-1}} - \mu_{A_t(S)} \right\rangle_{\mathcal{H}_\kappa}}{\left\| \mu_{Q_{t-1}} - \mu_{A_t(S)} \right\|^2_{\mathcal{H}_\kappa}}\right]_{[0,1]}. \tag{5}$$

Here, $[\cdot]_{[0,1]}$ clips to $[0, 1]$; we set $\lambda_t^\star(S) = 0$ when the denominator is zero. This closed form avoids iterative optimization of the mixture coefficient during candidate evaluation. The optimal mixture is used only for anchor evaluation; Sec. 3.4 constructs the final features and next state.

Greedy anchor selection. Starting from $S = \varnothing$, we add the unselected candidate that minimizes $\mathcal{L}_t(S \cup \{i\})$:

$$i^\star \in \arg\min_{i \notin S} \mathcal{L}_t(S \cup \{i\}), \qquad S \leftarrow S \cup \{i^\star\}. \tag{6}$$

Each candidate is evaluated jointly with the anchors already in S, rather than assigned a fixed score independently of the current selection. After K_t steps, we obtain S_t. Thus, λ is optimized per trial set, whereas anchors are constructed greedily. For the first frame, we set $\lambda = 0$ and match $A_t(S)$ to P_t without a state. All terms in Eqs. (4)–(5) can be evaluated through weighted sums of pairwise kernel values, without explicitly constructing RKHS embeddings. Within a frame, scene- and state-only statistics remain fixed, while sums involving selected anchors admit incremental updates as S grows. Appendices A.2–A.4 give the discrete MMD formulas, the derivation of Eq. (5), and cached candidate evaluation.

3.4 Visual State Construction

Mass-Preserving Aggregation. Unselected tokens may still contain information not fully represented by the state and selected anchors. We therefore aggregate their features and attention mass without increasing the token budget. Each token is assigned to its most similar anchor, $\pi_t(i) = \arg\max_{s \in S_t} \hat{x}_{t,i}^\top \hat{x}_{t,s}$, forming $\mathcal{C}_s = \{i : \pi_t(i) = s\}$; each anchor is assigned to itself.

Each anchor receives its cluster's total attention mass b_s. Inspired by FastVID's anchor-centric aggregation and UniComp's equal-weight neighbor fusion (Shen et al., 2025; Yuan et al., 2026), we combine the anchor direction with the normalized cluster mean to obtain d_s:

$$b_s = \sum_{i \in \mathcal{C}_s} a_{t,i}, \qquad \sum_{s \in S_t} b_s = 1, \tag{7}$$

$$\hat{c}_s = \nu\left(\frac{1}{|\mathcal{C}_s|} \sum_{i \in \mathcal{C}_s} \hat{x}_{t,i}\right), \qquad d_s = \nu\left(\frac{1}{2}\hat{x}_{t,s} + \frac{1}{2}\hat{c}_s\right), \tag{8}$$

where $\nu(z) = z/\|z\|_2$. Mass transfer preserves the attention mass assigned to each cluster. Feature fusion incorporates information from the assigned tokens while retaining an explicit contribution from the selected anchor, rather than replacing it with the cluster centroid alone.

For the LLM input, we restore feature scale by averaging the original anchor norm and the mean token norm within its cluster, $\bar{\rho}_s = |\mathcal{C}_s|^{-1} \sum_{i \in \mathcal{C}_s} \rho_{t,i}$:

$$v_{t,s}^{\text{out}} = \left(\frac{1}{2}\rho_{t,s} + \frac{1}{2}\bar{\rho}_s\right) d_s. \tag{9}$$

For the next visual state, we retain the directions $z_s = \nu(v_{t,s}^{\text{out}}) = d_s$ and aggregated attention masses b_s; norm restoration is used only for the LLM-input branch:

$$\boxed{Q_t = \sum_{s \in S_t} b_s \delta_{z_s}}. \tag{10}$$

The LLM receives the K_t vectors $v_{t,s}^{\text{out}}$; masses b_s only weight the next state's kernel mean embedding. Although Q_t contains current-frame anchors, they were selected conditionally on Q_{t-1}, linking successive selections. This state is a local reference, not a cumulative memory. The LLM receives compressed features from all frames in their original temporal and spatial order, while the conditioning state remains bounded by the per-frame budget.

4 Experiments

4.1 Experimental Settings

Benchmarks. We evaluate on five widely used video understanding benchmarks: **MVBench** (Li et al., 2024) and **TempCompass** (Liu et al., 2024) for fine-grained temporal understanding, **VideoMME** (Fu et al., 2025) for comprehensive video understanding, and **LongVideoBench** (Wu et al., 2024) and **LVBench** (Wang et al., 2025) for long-form video understanding.

Compared Baselines. We compare C^{2}V with four state-of-the-art training-free visual token compressors: **DivPrune** (Alvar et al., 2025), **VisionZip** (Yang et al., 2025), **FastVID** (Shen et al., 2025), and **FlashVID** (Fan et al., 2026).

Implementation Details. We evaluate **Qwen3-VL-8B** (Bai et al., 2025a), **InternVideo3-8B** (Yan et al., 2026), and **Qwen2.5-VL-7B** (Bai et al., 2025b). Unless otherwise noted, methods on the same backbone share its native sampling and preprocessing pipeline with a 32-frame sampling budget. Before-LLM retention $R \in \{10\%, 15\%, 20\%, 25\%\}$ is the visual-token count at the LLM input relative to the uncompressed sequence. We fix $\delta = 0.20$ across all settings.

FlashVID† denotes the before-LLM component with inner pruning disabled; FlashVID denotes the full hybrid. Table 1 compares before-LLM compression. In Table 3, FastVID, FlashVID, and C^{2}V share the same inner-LLM pruning configuration and matched LLM-input and layer-average visual-token counts. We use NVIDIA RTX A5000 24GB GPUs and LMMs-Eval (Zhang et al., 2025). Appendices B and C detail evaluation protocols, baselines, and backbone-specific implementation.

4.2 Main Results

Table 1 shows that C^{2}V achieves the highest five-benchmark average accuracy in all 12 model–retention settings.

Results on Qwen3-VL-8B. Compared with FlashVID†, the strongest baseline on this backbone, C^{2}V improves average accuracy by 0.7 percentage points at 25% retention and 1.7 at 10%, where it preserves 94.1% of the uncompressed average accuracy. At 15%, it reaches comparable average accuracy to FlashVID† at 25% (both reported as 58.8%) with 40% fewer LLM-input visual tokens.

Results on InternVideo3-8B. C^{2}V leads on MVBench and VideoMME at every tested retention ratio (Table 1). At 10%, it achieves 57.2% average accuracy, exceeding FlashVID† by 0.7 percentage points and preserving 94.2% of the uncompressed average accuracy.

Results on Qwen2.5-VL-7B. C^{2}V exceeds the strongest baseline at each retention ratio by 0.2–0.6 percentage points in average accuracy, reaching 57.8% at 25% retention (97.2% of the uncompressed average accuracy). Improvements vary across benchmarks, with several LongVideoBench settings favoring other methods.

Fixed-budget frame scaling on Qwen3-VL-8B. Table 2 tests whether additional sampled frames improve understanding at a fixed LLM visual-token budget, comparing 64 frames at 25% retention and 128 at 12.5% against the uncompressed 16-frame baseline. C^{2}V leads on all three benchmarks and every VideoMME duration subset, achieving three-benchmark averages of 55.9% and 56.5%, respectively—1.1 and 1.4 points above FlashVID†. At 128 frames, it improves the average by 9.3% over the 16-frame baseline and reaches 58.7% on VideoMME-Long, 3.4 points above FlashVID†. Within each setting, methods share the frame count and retention ratio, so the gains demonstrate more effective use of additional observations at the same LLM visual-token budget (Appendix C.3).

Hybrid compression on Qwen3-VL-8B. With common inner-LLM pruning and matched LLM-input and layer-average visual-token budgets, C^{2}V achieves the highest five-benchmark average accuracy at all four settings, exceeding FlashVID by 0.7–1.4 percentage points (Table 3). In a separate comparison at matched layer-average retention, before-LLM C^{2}V (Table 1) exceeds hybrid FlashVID by 0.2–0.7 points despite the latter's 25% larger visual-token input.

4.3 Ablation Studies

We conduct ablations on InternVideo3-8B at 10% retention, avoiding Qwen3-VL's additional DeepStack injection into early LLM layers. Mean accuracy is reported over MVBench (MVB), VideoMME (VMME), and LongVideoBench (LVB), one benchmark per category (Sec. 4.1).

Table 5 isolates the stage-wise components; Appendix C.4 defines the variants. Uniform scene masses use $1/N$ in P_t, while retaining the original attention weights for cluster-mass aggregation in Eq. (7). The no-state variants replace state-conditioned selection by attention top-K or by MMD with $\lambda = 0$, respectively. Uniform state masses replace b_s by $1/K_t$ in Q_t. The no-fusion variant uses original anchor features, omitting direction fusion and norm averaging while retaining cluster-mass aggregation.

Table 1: Before-LLM compression with 32 sampled frames. Score is the five-benchmark mean accuracy; Rel. Acc. (%) is $100 \times \mathrm{Score}/\mathrm{Score}_{\mathrm{Vanilla}}$ within each backbone. Bold marks the best unrounded result per backbone–retention setting.

Backbone	Method	Retention Ratio R	Fine-grained Temporal Understanding		Comprehensive Video Understanding	Long-form Video Understanding		Avg.	
			MVBench	TempCompass	VideoMME	LongVideoBench	LVBench	Score	Rel. Acc. (%)
Qwen3-VL-8B	Vanilla	*100%*	68.4	74.1	64.5	59.5	39.4	61.2	100.0
	DivPrune	*10%*	62.2	67.7	57.7	55.6	32.0	55.0	90.0
	VisionZip	*10%*	60.9	66.8	56.7	54.8	34.4	54.7	89.5
	FastVID	*10%*	61.8	65.8	57.7	**56.2**	35.1	55.3	90.5
	FlashVID†	*10%*	64.2	67.0	57.9	54.6	35.6	55.9	91.4
	C²V	*10%*	**66.2**	**70.1**	**59.3**	55.5	**36.7**	**57.6**	**94.1**
	DivPrune	*15%*	64.0	68.6	58.9	55.3	32.4	55.8	91.3
	VisionZip	*15%*	64.5	67.5	58.9	56.3	35.8	56.6	92.6
	FastVID	*15%*	64.3	67.2	59.6	57.1	36.3	56.9	93.0
	FlashVID†	*15%*	66.0	69.3	59.9	55.3	36.5	57.4	93.8
	C²V	*15%*	**67.6**	**70.9**	**60.5**	**57.1**	**37.6**	**58.8**	**96.1**
	DivPrune	*20%*	64.7	69.2	59.8	55.6	33.8	56.6	92.6
	VisionZip	*20%*	66.0	68.5	60.4	56.6	36.9	57.7	94.3
	FastVID	*20%*	65.2	67.4	60.4	57.3	37.2	57.5	94.0
	FlashVID†	*20%*	66.8	70.5	**61.2**	56.3	36.2	58.2	95.2
	C²V	*20%*	**68.3**	**71.5**	61.0	**57.8**	**37.8**	**59.3**	**96.9**
	DivPrune	*25%*	65.6	70.5	60.7	56.8	33.5	57.4	93.9
	VisionZip	*25%*	66.1	69.6	61.3	57.1	36.4	58.1	95.0
	FastVID	*25%*	65.9	68.4	61.0	**58.6**	**38.0**	58.4	95.5
	FlashVID†	*25%*	67.1	71.4	61.7	57.4	36.6	58.8	96.2
	C²V	*25%*	**68.2**	**71.8**	**62.1**	58.0	37.4	**59.5**	**97.3**
InternVideo3-8B	Vanilla	*100%*	72.2	72.8	64.1	55.3	39.4	60.7	100.0
	DivPrune	*10%*	64.2	66.9	57.6	50.6	35.4	54.9	90.4
	VisionZip	*10%*	62.4	68.0	57.9	49.2	34.3	54.4	89.5
	FastVID	*10%*	67.4	68.6	57.6	51.7	35.4	56.1	92.4
	FlashVID†	*10%*	68.7	68.9	57.7	**51.8**	35.4	56.5	93.0
	C²V	*10%*	**69.6**	**69.6**	**59.6**	51.5	**35.8**	**57.2**	**94.2**
	DivPrune	*15%*	66.5	68.2	59.4	51.5	35.8	56.3	92.6
	VisionZip	*15%*	68.7	69.4	59.6	51.7	**36.3**	57.1	94.0
	FastVID	*15%*	69.3	**70.2**	59.3	53.1	36.0	57.6	94.8
	FlashVID†	*15%*	70.3	70.1	58.1	**53.3**	35.3	57.4	94.5
	C²V	*15%*	**71.4**	69.8	**60.1**	52.0	36.2	**57.9**	**95.3**
	DivPrune	*20%*	67.8	69.2	60.3	51.6	36.3	57.0	93.9
	VisionZip	*20%*	70.1	70.3	60.3	53.0	**36.7**	58.1	95.6
	FastVID	*20%*	70.2	70.5	59.8	52.7	**36.7**	58.0	95.4
	FlashVID†	*20%*	70.8	70.8	59.0	53.5	35.6	57.9	95.4
	C²V	*20%*	**71.7**	**71.2**	**60.8**	**53.5**	36.3	**58.7**	**96.7**
	DivPrune	*25%*	69.3	70.5	60.8	51.8	36.8	57.8	95.2
	VisionZip	*25%*	71.6	71.3	61.4	**53.9**	36.7	59.0	97.1
	FastVID	*25%*	70.6	71.0	60.9	53.5	**37.3**	58.7	96.6
	FlashVID†	*25%*	71.7	70.6	60.2	53.3	37.0	58.5	96.4
	C²V	*25%*	**72.0**	**71.3**	**61.8**	53.5	36.7	**59.1**	**97.2**
Qwen2.5-VL-7B	Vanilla	*100%*	67.6	60.9	71.9	58.6	38.2	59.4	100.0
	DivPrune	*10%*	61.8	53.3	**64.7**	53.8	33.4	53.4	89.9
	VisionZip	*10%*	62.1	55.6	63.4	53.8	34.2	53.8	90.5
	FastVID	*10%*	62.3	55.3	64.5	54.8	34.3	54.2	91.3
	FlashVID†	*10%*	62.8	56.4	63.2	54.4	**35.5**	54.5	91.6
	C²V	*10%*	**63.5**	**56.7**	**64.7**	**55.3**	35.2	**55.1**	**92.8**
	DivPrune	*15%*	63.2	55.2	65.8	55.3	34.9	54.9	92.4
	VisionZip	*15%*	64.3	57.5	65.9	55.3	34.7	55.5	93.5
	FastVID	*15%*	63.7	57.3	66.1	**56.0**	35.1	55.7	93.7
	FlashVID†	*15%*	64.5	57.3	66.0	55.9	35.8	55.9	94.1
	C²V	*15%*	**64.7**	**57.7**	**66.5**	55.9	**36.0**	**56.1**	**94.5**
	DivPrune	*20%*	63.8	56.1	67.2	55.3	35.8	55.6	93.6
	VisionZip	*20%*	65.6	**58.5**	67.3	56.4	36.0	56.8	95.5
	FastVID	*20%*	64.2	57.5	67.7	**57.1**	35.3	56.3	94.8
	FlashVID†	*20%*	**65.8**	58.2	67.1	56.9	36.0	56.8	95.6
	C²V	*20%*	65.7	58.4	**67.9**	56.7	**36.5**	**57.0**	**96.0**
	DivPrune	*25%*	64.7	57.3	68.1	56.3	35.0	56.3	94.7
	VisionZip	*25%*	65.9	**59.0**	68.2	57.2	36.4	57.3	96.5
	FastVID	*25%*	64.1	58.3	68.4	**57.8**	36.3	57.0	95.9
	FlashVID†	*25%*	66.2	58.5	68.2	56.6	36.1	57.1	96.1
	C²V	*25%*	**66.4**	58.7	**68.6**	57.5	**37.6**	**57.8**	**97.2**

Table 2: Fixed-budget frame scaling on Qwen3-VL-8B. Both settings match the uncompressed 16-frame baseline's visual-token budget. Score averages VideoMME (Overall), LongVideoBench, and LVBench; Rel. Acc. is relative to this baseline.

Method	Sampled frames	Retention Ratio R	VideoMME				LongVideoBench	LVBench	Avg.	
			Short	Medium	Long	Overall			Score	Rel. Acc. (%)
Vanilla	16	*100%*	70.2	57.4	52.3	60.0	57.9	37.4	51.8	100.0
FastVID	64	*25%*	74.0	61.2	53.9	63.0	60.1	38.7	53.9	104.2
FlashVID†			76.6	62.1	54.9	64.5	59.4	40.6	54.8	106.0
C²V			**77.4**	**64.6**	**56.1**	**66.0**	**60.9**	**40.9**	**55.9**	**108.1**
FastVID	128	*12.5%*	75.2	61.3	55.2	63.9	59.9	40.1	54.6	105.6
FlashVID†			76.0	64.2	55.3	65.2	58.8	41.3	55.1	106.4
C²V			**77.1**	**65.1**	**58.7**	**67.0**	**60.6**	**42.1**	**56.5**	**109.3**

Table 3: Hybrid compression on Qwen3-VL-8B using FlashVID's inner-LLM pruning, retaining the top 10% of visual tokens by LLM attention at layer 28. R ("Avg.") is the layer-average visual-token count relative to the uncompressed count; before-LLM retention is $1.25R$.

Method	Retention Ratio R		MVBench	TempCompass	VideoMME	LongVideoBench	LVBench	Avg.	
	Before LLM	Avg.						Score	Rel. Acc. (%)
Vanilla	*100%*	*100%*	68.4	74.1	64.5	59.5	39.4	61.2	100.0
FastVID	*12.5%*	*10%*	63.4	66.2	59.2	**56.6**	35.7	56.2	91.9
FlashVID			65.3	68.3	59.4	54.9	36.7	56.9	93.1
C²V			**67.1**	**71.0**	**59.5**	56.2	**37.6**	**58.3**	**95.3**
FastVID	*18.75%*	*15%*	65.3	67.2	60.6	**57.4**	37.4	57.6	94.1
FlashVID			66.2	70.2	**61.1**	56.4	36.8	58.1	95.1
C²V			**67.9**	**71.9**	60.6	**57.4**	**38.2**	**59.2**	**96.8**
FastVID	*25%*	*20%*	65.8	68.4	60.9	**58.7**	**37.8**	58.3	95.4
FlashVID			67.0	71.1	**61.9**	57.5	36.9	58.9	96.3
C²V			**68.3**	**71.8**	61.9	58.6	37.5	**59.6**	**97.5**
FastVID	*31.25%*	*25%*	66.3	69.4	62.6	58.4	37.8	58.9	96.3
FlashVID			67.5	71.6	62.0	58.5	37.0	59.3	97.0
C²V			**68.3**	**72.8**	**62.6**	**58.6**	**38.2**	**60.1**	**98.2**

Kernel scale. Across five kernel scales, average accuracy ranges from 59.7% to 60.3% (Table 4). The default $\delta = 0.2$ leads on all three benchmarks; adjacent scales $\delta = 0.1$ and 0.3 remain within 0.4 percentage points of its average accuracy.

Component contributions. Uniform scene masses lower average accuracy by 1.2 percentage points (Table 5), favoring attention-weighted scene modeling. Removing the state while retaining MMD selection and aggregation lowers it by 1.1 points; attention top-K selection lowers it by 1.2 points. Fixing $\lambda = 0.5$ improves the average over no-state MMD (59.5% vs. 59.2%) but remains below C²V on all three benchmarks, separating the benefit of state use from candidate-adaptive mixing. Removing feature fusion lowers average accuracy by 0.9 points despite retaining cluster-mass aggregation. Uniform state masses lower it by 0.5 points. For fixed anchors and clusters, changing b_s leaves Eq. (9) unchanged and affects subsequent selection through Q_t, supporting cluster attention mass in the visual state.

Kernel choice. Among the tested kernels (Appendix A.5), characteristic choices yield 59.7–60.3% average accuracy, versus 59.4–59.6% for linear and degree-2 polynomial kernels (Table 6). Laplacian leads at 60.3%, followed by RBF (60.0%) and IMQ (59.9%), suggesting that strong performance is not specific to the Laplacian kernel.

4.4 Analysis

Video-level feature coverage. We compare the final outputs of C²V, VisionZip, FastVID, and FlashVID† on 900 VideoMME videos using Qwen3-VL-8B, 32 sampled frames, and 10% retention. For each video, we evaluate source-feature reconstruction using a rank-r subspace derived jointly from the final aggregated outputs across the video, relative to the optimal rank-r reconstruction from

Table 4: Effect of the Laplacian kernel scale δ.

δ	MVB	VMME	LVB	Avg.
0.1	69.5	59.3	50.9	59.9
0.2	**69.6**	**59.6**	**51.5**	**60.3**
0.3	69.5	59.4	51.2	60.0
0.4	69.1	58.9	51.0	59.7
0.5	69.0	59.1	51.0	59.7

Table 5: Component ablations of C^{2}V.

Variant	MVB	VMME	LVB	Avg.
Scene Distribution Construction:				
Uniform scene masses	67.2	58.8	51.2	59.1
State-Conditioned Scene Completion:				
w/o state (top-K attn.)	68.3	59.1	50.0	59.1
w/o state (MMD)	68.3	58.4	51.0	59.2
fixed mixture ($\lambda = 0.5$)	69.4	58.3	50.9	59.5
Visual State Construction:				
Uniform state masses	69.5	**59.6**	50.3	59.8
w/o feature fusion	69.5	57.6	51.1	59.4
C^{2}V (Ours)	**69.6**	**59.6**	**51.5**	**60.3**

Table 6: Comparison of kernel functions for MMD-based scene completion, grouped by characteristic property.

Kernel	MVB	VMME	LVB	Avg.
Non-characteristic kernels:				
Linear	68.5	58.7	51.1	59.4
Polynomial-2	68.9	59.0	50.9	59.6
Characteristic kernels:				
RBF	**69.9**	59.1	50.9	60.0
IMQ	69.4	59.0	**51.5**	59.9
Energy ($q = 1$)	69.1	58.6	51.3	59.7
Laplacian (Ours)	69.6	**59.6**	**51.5**	**60.3**

Table 7: Prefilling latency comparison on Qwen3-VL-8B. R denotes the retention ratio.

Method	R	Prefilling Time (ms)			Avg.
		Comp.	LLM Forward	Total	
FastVID	*25%*	34.5	415.8	450.3	58.4
FlashVID†	*20%*	71.6	365.3	436.9	58.2
C^{2}V	*15%*	**29.9**	**299.0**	**328.9**	**58.8**

all source tokens under the same weighting. We average these relative scores across videos. Source-token features x_i receive uniform, final-layer vision-encoder attention, or feature-uniqueness scores $s_i^{\text{uniq}} = 1 - \max_{j \neq i} \cos(x_i, x_j)$, where j ranges over the other source tokens in the video. Weights are normalized per frame to give each frame equal total mass and are shared across compressors. Each compressor's output subspace is also fixed across the three weightings. This post-hoc metric evaluates collective source-feature reconstruction using video-level subspaces of final aggregated outputs, not the frame-level MMD objective. Appendix D defines the metric and its scope; Appendix F visualizes selected anchor locations.

At every tested rank ($r = 32, 64, 128, 256$) and under all three weightings, C^{2}V exceeds the strongest baseline by 1.6–4.1 percentage points (Figure 1). The advantage extends beyond attention-weighted evaluation to uniform weighting and feature-uniqueness weighting, supporting improved collective feature representation by the complete pipeline.

Efficiency analysis. Table 7 compares Qwen3-VL-8B configurations at similar five-benchmark average accuracy rather than equal retention. Latencies are averaged over 2,700 VideoMME questions, excluding shared vision encoding. At 15% retention, C^{2}V achieves 58.8% average accuracy, exceeding FastVID at 25% (58.4%) and FlashVID† at 20% (58.2%) with 40% and 25% fewer visual tokens, respectively. Its combined compression and LLM prefilling latency is 328.9 ms, 27.0% and 24.7% lower than these baselines, respectively, with the lowest compression latency (29.9 ms) among these configurations. Additional compressor latencies and complexity bounds appear in Appendix E.

5 Conclusion

We presented C^{2}V, a training-free, before-LLM compressor combining state-conditioned MMD selection with mass-preserving aggregation. With one fixed method-specific hyperparameter, it leads the compared methods in five-benchmark average accuracy across all 12 model–retention settings. Matched-budget analysis shows higher video-level feature coverage. At the evaluated Qwen3-VL operating points, it achieves higher average accuracy with 24.7–27.0% lower combined compression and LLM prefilling latency. These results favor complementarity to retained representations over frame-wise importance alone.

Ethics statement

This work studies training-free video token compression using existing benchmarks and pretrained Video Large Language Models. It does not involve new participant recruitment or collection of personal data. More efficient video analysis may improve accessibility, but may also lower the cost of privacy-invasive applications. Token compression can discard task-relevant information, and improved average benchmark accuracy does not establish fairness or suitability for safety-critical use. Deployment should respect the original model and dataset terms and include application-specific privacy, safety, and reliability evaluation.

Reproducibility statement

Section 3 describes the complete C^2V formulation and algorithm, and Sec. 4 specifies the evaluation settings and protocols. The Appendix provides additional derivations, implementation details, feature-coverage definitions, and latency measurement settings. An anonymized implementation, configuration files, and evaluation scripts are included in the supplementary material.

AI Use Statement

Generative AI tools were used to assist with writing and polishing the manuscript, including improvements to clarity and grammar. The authors reviewed and verified the final manuscript.

References

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

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

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

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

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Jim Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2e163450c1ae3167832971e6da29f38d-Paper-Conference.pdf.

Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pp. 109–116, Arlington, Virginia, USA, 2010. AUAI Press. ISBN 9780974903965.

Raaz Dwivedi and Lester Mackey. Kernel thinning. In Mikhail Belkin and Samory Kpotufe (eds.), *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 1753–1753. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/dwivedi21a.html.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 98730–98759, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a0698477f358a6adeab2c2b44f109e5e-Paper-Conference.pdf.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pp. 396–414, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-20082-3. doi: 10.1007/978-3-031-20083-0_24. URL https://doi.org/10.1007/978-3-031-20083-0_24.

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, June 2025.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Jiameng Li, Minye Wu, Jiezhang Cao, Aleksei Tiulpin, and Matthew B Blaschko. Echoprune: Interpreting redundancy as temporal echoes for efficient videollms. *arXiv preprint arXiv:2605.10050*, 2026.

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

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BjyvwnXXVn_.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024.

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

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

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5): 2263–2291, 2013. doi: 10.1214/13-AOS1140.

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

Kele Shao, Keda TAO, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. Holitom: Holistic token merging for fast video large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6hvaQTKkpF.

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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 123553–123581. Curran Associates, Inc., 2025. doi: 10.52202/085713-4118. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b2e63e36c57e153b9015fece2352a9f9-Paper-Conference.pdf.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22958–22967. IEEE, 2025.

Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2092–2101. IEEE, 2023.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pp. 453–470. Springer, 2024.

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

Ziang Yan, Sheng Xia, Jiashuo Yu, Yue Wu, Tianxiang Jiang, Songze Li, Kanghui Tian, Yicheng Xu, Yinan He, Kai Chen, et al. InternVideo3: Agentify foundation models with multimodal contextual reasoning. *arXiv preprint arXiv:2606.12195*, 2026.

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

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

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-eval: Reality check on the evaluation of large multimodal models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.51. URL https://aclanthology.org/2025.findings-naacl.51/.

A Additional Method Details

A.1 Overall Algorithm

Algorithm A.1 summarizes C^{2}V using the notation of Sec. 3.1. The state is reset for each video and replaced by the current aggregated distribution, not by the mixture used for candidate evaluation. We write $\nu(z) = z/\|z\|_2$.

Algorithm A.1 C^{2}V: recurrent visual token compression

Require: Features $\{V_t\}_{t=1}^T$, aligned attention $\{\alpha_t\}_{t=1}^T$, retention R, kernel scale $\delta = 0.20$
Ensure: Compressed features $\mathcal{O}$ and original anchor positions

$Q_{\text{prev}} \leftarrow \varnothing$, $\mathcal{O} \leftarrow [\,]$
Allocate $K_1, \dots, K_T$ using Eq. (C.2)
for $t = 1, \dots, T$ **do**
  $\rho_{t,i} \leftarrow \|v_{t,i}\|_2$, $\hat{x}_{t,i} \leftarrow \nu(v_{t,i})$
  $a_{t,i} \leftarrow \alpha_{t,i} / \sum_j \alpha_{t,j}$; $P_t \leftarrow \sum_i a_{t,i} \delta_{\hat{x}_{t,i}}$
  Precompute kernel statistics; $S \leftarrow \varnothing$; initialize accumulators
  for $\ell = 1, \dots, K_t$ **do**
    for all $i \notin S$ **in parallel do**
      $A_i \leftarrow (|S| + 1)^{-1} \sum_{s \in S \cup \{i\}} \delta_{\hat{x}_{t,s}}$
      if $Q_{\text{prev}} = \varnothing$ **then**
        $\lambda_i \leftarrow 0$; $J_i \leftarrow \text{MMD}^2_\kappa(P_t, A_i)$
      else
        $\lambda_i \leftarrow \lambda^\star(S \cup \{i\})$ using Eq. (A.5)
        $J_i \leftarrow \text{MMD}^2_\kappa(P_t, \lambda_i Q_{\text{prev}} + (1 - \lambda_i) A_i)$
      end if
    end for
    $j \leftarrow \arg\min_{i \notin S} J_i$
    Update accumulators by Eq. (A.8); $S \leftarrow S \cup \{j\}$
  end for
  Form cosine-nearest-anchor clusters $\mathcal{C}_s$, assigning each anchor to itself
  for all $s \in S$ **do**
    $b_s \leftarrow \sum_{i \in \mathcal{C}_s} a_{t,i}$, $\hat{c}_s \leftarrow \nu\big(|\mathcal{C}_s|^{-1} \sum_{i \in \mathcal{C}_s} \hat{x}_{t,i}\big)$
    $d_s \leftarrow \nu((\hat{x}_{t,s} + \hat{c}_s)/2)$, $\bar{\rho}_s \leftarrow |\mathcal{C}_s|^{-1} \sum_{i \in \mathcal{C}_s} \rho_{t,i}$
    $v^{\text{out}}_{t,s} \leftarrow (\rho_{t,s} + \bar{\rho}_s) d_s / 2$
  end for
  Append $v^{\text{out}}_{t,s}$ to $\mathcal{O}$ in source order, preserving anchor positions
  $Q_{\text{prev}} \leftarrow \sum_{s \in S} b_s \delta_{\nu(v^{\text{out}}_{t,s})}$
end for
return $\mathcal{O}$ and original anchor positions

A.2 Weighted Discrete MMD

Suppressing the temporal index, let $P = \sum_{i=1}^{N} a_i \delta_{\hat{x}_i}$, $Q = \sum_{j=1}^{M} b_j \delta_{z_j}$, and $A(S) = m^{-1} \sum_{s \in S} \delta_{\hat{x}_s}$, where $m = |S| > 0$. The masses of each distribution sum to one. Define

$$K_{iu} = \kappa(\hat{x}_i, \hat{x}_u), \qquad C_{ij} = \kappa(\hat{x}_i, z_j), \qquad G_{jv} = \kappa(z_j, z_v). \tag{A.1}$$

The six mean-embedding inner products are

$$\begin{aligned} &s_{PP} = a^\top K a, && s_{QQ} = b^\top G b, && s_{PQ} = a^\top C b, \\ &s_{PA} = \frac{1}{m} \sum_{s \in S} (Ka)_s, && s_{QA} = \frac{1}{m} \sum_{s \in S} (Cb)_s, && s_{AA} = \frac{1}{m^2} \sum_{s,u \in S} K_{su}. \end{aligned} \tag{A.2}$$

We evaluate MMD directly between these finite weighted measures, including diagonal kernel terms (Gretton et al., 2012). The mixture objective becomes

$$\begin{aligned} J_S(\lambda) &= \mathrm{MMD}_\kappa^2\big(P, \lambda Q + (1-\lambda) A(S)\big) \\ &= s_{PP} + \lambda^2 s_{QQ} + (1-\lambda)^2 s_{AA} + 2\lambda(1-\lambda) s_{QA} \\ &\quad - 2\lambda s_{PQ} - 2(1-\lambda) s_{PA}. \end{aligned} \tag{A.3}$$

A.3 Optimal Mixture Coefficient

Set

$$\begin{aligned} e &= s_{PP} + s_{AA} - 2 s_{PA}, \\ d &= s_{QQ} + s_{AA} - 2 s_{QA}, \\ c &= s_{PQ} - s_{PA} - s_{QA} + s_{AA}. \end{aligned} \tag{A.4}$$

Then $J_S(\lambda) = e - 2c\lambda + d\lambda^2$, with derivative $J_S'(\lambda) = -2c + 2d\lambda$. Since $d = \|\mu_Q - \mu_{A(S)}\|_{\mathcal{H}_\kappa}^2 \geq 0$, the constrained minimum is

$$\lambda^\star(S) = \begin{cases} \mathrm{clip}_{[0,1]}(c/d), & d > 0, \\ 0, & d = 0. \end{cases} \tag{A.5}$$

The identity $c = \langle \mu_P - \mu_{A(S)}, \mu_Q - \mu_{A(S)} \rangle_{\mathcal{H}_\kappa}$ recovers the main text's projection formula. If $d = 0$, the two endpoints coincide and the objective is independent of λ. For the first frame, there is no preceding state, so $\lambda = 0$ and the score is e.

A.4 Efficient Greedy Candidate Evaluation

Within each frame, cache $p = Ka$ and $s_{PP} = a^\top p$, together with $h = Cb$, $s_{PQ} = a^\top h$, and s_{QQ} when a preceding state is available. For the selected set S, maintain the accumulators

$$q_i = \sum_{s \in S} K_{is}, \quad L = \sum_{s,u \in S} K_{su}, \quad U = \sum_{s \in S} p_s, \quad V = \sum_{s \in S} h_s, \tag{A.6}$$

initialized to zero. For a trial set $S \cup \{i\}$ of size $n = |S| + 1$,

$$s_{AA}^{(i)} = \frac{L + 2q_i + K_{ii}}{n^2}, \qquad s_{PA}^{(i)} = \frac{U + p_i}{n}, \qquad s_{QA}^{(i)} = \frac{V + h_i}{n}. \tag{A.7}$$

These statistics give $c_i, d_i, \lambda_i^\star$, and J_i for all unused candidates. After choosing j, update

$$\begin{aligned} &L \leftarrow L + 2q_j + K_{jj}, && U \leftarrow U + p_j, \\ &V \leftarrow V + h_j, && q \leftarrow q + K_{:,j}, \end{aligned} \tag{A.8}$$

using the pre-update q_j. With no preceding state, only $s_{PP} + s_{AA}^{(i)} - 2 s_{PA}^{(i)}$ is required. The greedy updates cost $O(K_t N)$ after kernel construction, excluding aggregation; the current dense Gram matrix occupies $O(N^2)$ memory.

A.5 Kernel Definitions and Settings

For unit directions, write $c_{xy} = \hat{x}^\top \hat{y}$ and $d_c = 1 - c_{xy}$. Table A.1 gives the evaluated kernels. Laplacian, RBF, and IMQ share a half-response at $d_c = \delta$: with $\delta = 0.20$, cosine similarity 0.8 yields kernel similarity 0.5. We keep this default across backbones, benchmarks, and retention ratios; the Laplacian sensitivity analysis varies $\delta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. The energy kernel is the Euclidean distance-induced kernel with $q = 1$, restricted to unit-norm features, and is characteristic on this domain (Sejdinovic et al., 2013).

Table A.1: Kernel functions and settings used in the ablation.

Kernel	$\kappa(\hat{x}, \hat{y})$	Setting
Laplacian	$2^{-\sqrt{d_c/\delta}}$	$\delta = 0.20$
RBF	$2^{-d_c/\delta}$	$\delta = 0.20$
IMQ	$(1 + 3d_c/\delta)^{-1/2}$	$\delta = 0.20$
Energy	$1 - \frac{1}{2}\sqrt{2d_c}$	$q = 1$
Linear	c_{xy}	No scale or bias
Polynomial-2	$((1 + c_{xy})/2)^2$	Degree 2

B Benchmarks and Compared Methods

B.1 Benchmarks and Evaluation Protocols

Benchmark scope. MVBench (Li et al., 2024) and TempCompass (Liu et al., 2024) assess fine-grained temporal understanding, including action, motion, and event order. VideoMME (Fu et al., 2025) covers diverse video content across short, medium, and long durations. LongVideoBench (Wu et al., 2024) evaluates referring and reasoning over long video contexts, while LVBench (Wang et al., 2025) covers long-form recognition, retrieval, temporal grounding, and reasoning. Table B.1 specifies the multiple-choice evaluation scope used here.

Table B.1: Evaluation scope. Split names refer to the local dataset loaders; no model training is performed on these questions.

Benchmark	Loader split	Questions	Aggregation
MVBench	Train	4,000	Mean accuracy over 20 tasks
TempCompass	Test	1,580	Multiple-choice accuracy
VideoMME	Test	2,700	Overall and duration-group accuracy
LongVideoBench	Validation	1,337	Video-only QA accuracy
LVBench	Train	1,549	Question-level accuracy

Inputs and scoring. All evaluations use sampled RGB frames and question–option text, without separate subtitles or audio. VideoMME has 900 videos and 2,700 questions, evenly divided among the three duration groups. We use each benchmark adapter's native prompt, answer-extraction, and aggregation rules.

TempCompass scoring. We follow the official LMMs-Eval setup and evaluate the `multi-choice` configuration on the `test` split, reporting overall question-level accuracy across five temporal categories: Speed, Direction, Action, Order, and Attribute Change.

Reported scores. The main score averages five benchmark accuracies; frame scaling averages VideoMME (Overall), LongVideoBench, and LVBench. Relative accuracy is the ratio to the corresponding uncompressed mean, in percent. Means and relative accuracies are computed from unrounded benchmark accuracies before display rounding.

B.2 Baseline Methods and Configurations

We integrate reference operators with the native video backbones. The upstream revisions are `8f86b55` (VisionZip), `799e2d9` (DivPrune), `a40a109` (FastVID), and `983cce6` (FlashVID). Below, R denotes nominal before-LLM retention; Qwen refers to both Qwen3-VL and Qwen2.5-VL.

VisionZip. VisionZip (Yang et al., 2025) combines attention-selected dominant tokens with contextual representatives of the remaining tokens. We allocate 0.05 of the original token count to contextual tokens and $R - 0.05$ to dominant tokens. Routing uses spatially aligned vision-key features, and a contextual output adds the mean of its assigned non-representative tokens to the representative feature. The Qwen port calibrates the total budget; InternVideo3 rounds the two counts per frame. We use the training-free configuration.

DivPrune. DivPrune (Alvar et al., 2025) performs cosine-distance-based max–min diversity selection over the full visual sequence, without feature fusion. The Qwen port selects $\mathrm{round}(RTN)$ tokens; the InternVideo3 adapter passes the nominal retention to the reference operator.

FastVID. FastVID (Shen et al., 2025) combines dynamic segmentation, attention-based selection, and density-based merging. We use segmentation parameters $c = 8$, $\tau = 0.84$, and `ignore`$= 0.95$; DTM budget fraction $d = 0.40$; and merging parameters $p = 4$ and $\beta = 0.60$. The Qwen port takes the largest admissible integer allocation within the requested budget, without filling unused slots. InternVideo3 uses the reference operator at nominal R, with its attention signal computed from spatially pooled vision queries and keys.

FlashVID. FlashVID (Fan et al., 2026) combines attention–diversity selection and segmentation with tree-based temporal merging. Both ports enable segmentation and complementary segments, with $\alpha = 0.70$, temporal threshold 0.8, and segment threshold 0.9. The minimum segment count is four for Qwen and eight for InternVideo3. FlashVID† disables inner-LLM pruning; hybrid experiments use the common rule in Appendix C.3. Official implementations of MMG-Vid and EchoPrune were unavailable at the time of evaluation.

C Implementation and Ablation Details

C.1 Backbones and Evaluation Settings

Table C.1 gives the checkpoints and compression locations. The default input contains 32 sampled frames. Qwen uses linearly spaced frame indices with native pixel-budget settings `min_pixels=200704` and `max_pixels=501760`. InternVideo3 uses native 32-frame sampling and `pixels_per_frame=131072`. These are pixel-budget arguments used with native aspect-ratio-preserving preprocessing.

Table C.1: Backbone checkpoints and compression locations. The backend is the configured model-loading attention backend.

Backbone	Checkpoint and insertion point	Backend
Qwen3-VL-8B	`Qwen/Qwen3-VL-8B-Instruct`; after the native visual merger, before the LLM	`flash_attention_2`
Qwen2.5-VL-7B	`Qwen/Qwen2.5-VL-7B-Instruct`; after the native visual merger, before the LLM	`flash_attention_2`
InternVideo3-8B	`yanziang/InternVideo3-8B-Instruct`; after native video-feature extraction, before the decoder	`sdpa`

We use NVIDIA RTX A5000 24GB GPUs, BF16 model inference, and FP32 feature normalization and kernel statistics. C^{2}V's default outer-evaluation batch sizes are two for Qwen3-VL and one for

Qwen2.5-VL and InternVideo3; Qwen3-VL hybrid and frame-scaling experiments use batch size one. Generation is greedy, with at most 16 new tokens for MVBench, VideoMME, and LVBench, and 32 for LongVideoBench and TempCompass; native stopping rules are retained. Dependency versions and runnable configurations are provided in the supplementary code.

C.2 Attention and Positional Alignment

For C^{2}V on Qwen3-VL and InternVideo3, importance is the incoming attention of each key token in the last vision layer, averaged over heads and queries within the same frame. Averaging the four patches of each spatial merge group aligns the scores with the output features.

For Qwen2.5-VL, importance likewise comes from the final vision block's self-attention, not text or [CLS] attention. With RoPE-transformed queries and keys in frame t, we compute

$$\mathbf{A}_t^{(h)} = \mathrm{softmax}_{\mathrm{key}}\left(\frac{\mathbf{Q}_t^{(h)}\mathbf{K}_t^{(h)\top}}{\sqrt{d_h}}\right), \qquad \tilde{\alpha}_{t,j} = \frac{1}{HM}\sum_{h=1}^{H}\sum_{i=1}^{M}\big[\mathbf{A}_t^{(h)}\big]_{ij}. \tag{C.1}$$

Here H is the number of heads, M the token count per frame before spatial merging, and d_h the head dimension. We average scores within native 2×2 spatial merge groups, then undo the window permutation for both scores and features to preserve alignment. Scores are normalized to unit mass within each frame; attention is never averaged across frames.

Outputs retain their anchors' original positional IDs and source order; nonvisual text tokens are preserved. Qwen3-VL DeepStack features are gathered at the retained indices without additional fusion, and pruned positions receive no injection. The InternVideo3 checkpoint has no active DeepStack branch.

C.3 Token Budgets and Hybrid Pruning

Before-LLM allocation. For T frames at the compressor input, each with N tokens, C^{2}V allocates

$$B = \min(TN, \max(T, \lceil RTN \rceil)), \qquad K_t = \lfloor B/T \rfloor + \mathbf{1}\{t \le B \bmod T\}. \tag{C.2}$$

The total budget is rounded upward and distributed nearly uniformly, assigning any remainder to the earliest frames and retaining at least one anchor per frame.

Hybrid configuration. Qwen3-VL has 36 decoder layers. Pruning occurs after the first 28 layers, retaining 10% of visual tokens for the last eight layers according to head-averaged last-query attention from the preceding layer; text tokens are retained. For uncompressed count N_0, outer count M, and inner count $J = \lceil 0.10M \rceil$, the realized layer-average retention is

$$\bar{R}_{\mathrm{realized}} = \frac{28M + 8J}{36N_0}. \tag{C.3}$$

Ignoring integer rounding gives

$$\bar{R} = \frac{28 + 8(0.10)}{36} R_{\mathrm{outer}} = 0.8R_{\mathrm{outer}}, \qquad R_{\mathrm{outer}} = 1.25\bar{R}. \tag{C.4}$$

All hybrid compressors use this pruning rule.

Frame scaling. We compare 64- and 128-frame inputs at 25% and 12.5% retention, respectively, against an uncompressed 16-frame reference. All three Qwen3-VL settings use `min_pixels=200704`, `max_pixels=501760`, spatial patch size 16, and spatial merge size two. Native aspect-ratio-preserving resizing determines the spatial grid; we do not impose a single fixed $H \times W$ grid across videos. For a common spatial grid, let $N_0(F)$ denote the uncompressed visual-token count at the compressor input for F sampled frames. The nominal retained counts satisfy

$$N_0(16) = 0.25\, N_0(64) = 0.125\, N_0(128). \tag{C.5}$$

For each method, the nominal retention is adjusted for its integer allocation so that the realized retained-token count matches the 16-frame baseline budget within 0.2%.

C.4 Component Ablation Definitions

Table C.2 specifies the changes to the Laplacian C^{2}V reference. Anchor budgets, input processing, and positional handling are unchanged across variants.

Table C.2: Component ablations. Other construction steps follow the reference.

Variant	Change relative to C^{2}V
Uniform scene masses	Use uniform masses only in P_t for selection; cluster masses retain the original attention weights.
w/o state (MMD)	Set $\lambda = 0$ in every frame; retain MMD selection and aggregation.
Fixed mixture ($\lambda = 0.5$)	Fix $\lambda = 0.5$ during candidate evaluation when a preceding state is available; use $\lambda = 0$ otherwise. Retain greedy MMD selection and visual state construction.
w/o state (top-K attn.)	Select K_t anchors by original attention rather than state-conditioned MMD; retain aggregation.
Uniform state masses	Use $b_s = 1/K_t$ in Q_t; retain feature construction.
w/o feature fusion	Emit the original anchor feature $v_{t,s}^{\text{out}} = v_{t,s}$, removing both direction fusion and norm averaging; retain cluster assignment and attention-mass transfer.

D Video-Level Feature Coverage

We detail the feature-coverage protocol used in Figure 1 of the main paper and Figure D.1 below. Figure 1 evaluates 900 VideoMME videos using Qwen3-VL-8B, 32 sampled frames, and 10% retention. The compared methods are C^{2}V with the Laplacian kernel, VisionZip, FastVID, and FlashVID†. They share each video's source features and attention; each compressed representation includes the method's final aggregation outputs.

D.1 Representation and Relative Coverage

For a video, let $X \in \mathbb{R}^{M\times D}$ contain the source features and $Y \in \mathbb{R}^{L\times D}$ the final compressed features. We normalize only source rows, $u_i = x_i/\|x_i\|_2$, forming U; Y retains its output magnitudes. Neither matrix is mean-centered. Let $V_r(Y)$ contain up to r leading right singular vectors of Y. The per-token projection score is

$$q_i(Y, r) = \|u_i^\top V_r(Y)\|_2^2 = 1 - \|u_i - V_r(Y)V_r(Y)^\top u_i\|_2^2. \tag{D.1}$$

For shared nonnegative weights w summing to one and $W = \operatorname{diag}(w)$, define

$$C_{w,r}(Y) = \sum_i w_i q_i(Y, r) = \|\sqrt{W}\, U V_r(Y)\|_F^2. \tag{D.2}$$

The corresponding optimal rank-r reference is

$$C_{w,r}^* = \max_{V^\top V = I_r} \|\sqrt{W}\, U V\|_F^2 = \sum_{k=1}^{r} \sigma_k(\sqrt{W}\, U)^2. \tag{D.3}$$

Relative coverage and its video-level average are

$$R_{w,r}(Y) = 100\frac{C_{w,r}(Y)}{C_{w,r}^*}, \qquad \overline{R}_{w,r} = \frac{1}{n_v}\sum_{v=1}^{n_v} R_{w,r}^{(v)}. \tag{D.4}$$

Here n_v is the number of evaluated videos for that setting; the main paper's VideoMME analysis uses $n_v = 900$. The same weighted reference is used across methods for each video. The output basis is derived from raw Y and is shared across the three weightings; it is not fitted separately to each weighted objective.

D.2 Source-Token Weighting

Write $M = TN$ for T frames and N spatial tokens per frame. For each weighting, scores are normalized as

$$w_{t,j} = \frac{1}{T} \frac{s_{t,j}}{\sum_{k=1}^{N} s_{t,k}}, \qquad s_i^{\mathrm{unw}} = 1, \quad s_i^{\mathrm{attn}} = a_i, \quad s_i^{\mathrm{uniq}} = 1 - \max_{k \neq i} u_i^\top u_k. \tag{D.5}$$

Here a_i is source vision attention, and uniqueness searches all other source tokens in the video. Each frame receives total weight $1/T$, while the subspace is constructed jointly over the entire video.

D.3 Subspace Computation

Analysis uses FP32 features. We approximate the SVDs of Y and $\sqrt{W}\,U$ with randomized SVD, using 24 oversampling dimensions, three power iterations, and fixed video-level seeds. A single output decomposition up to rank 256 supplies the prefixes for $r \in \{32, 64, 128, 256\}$. If fewer than r usable output directions are available, we use those directions while keeping the source reference at the requested rank. The reference in Eq. (D.3) is estimated by the same SVD procedure.

D.4 Additional Coverage Results

Figure D.1 extends the analysis to VideoMME at 20% retention and to LVBench and TempCompass at 10% and 20%. C^{2}V achieves higher relative coverage than the compared methods at every displayed rank under all three source-token weightings. The ordering persists at 20% retention and on the two additional benchmarks, extending the collective feature-representation results beyond the main paper's VideoMME setting.

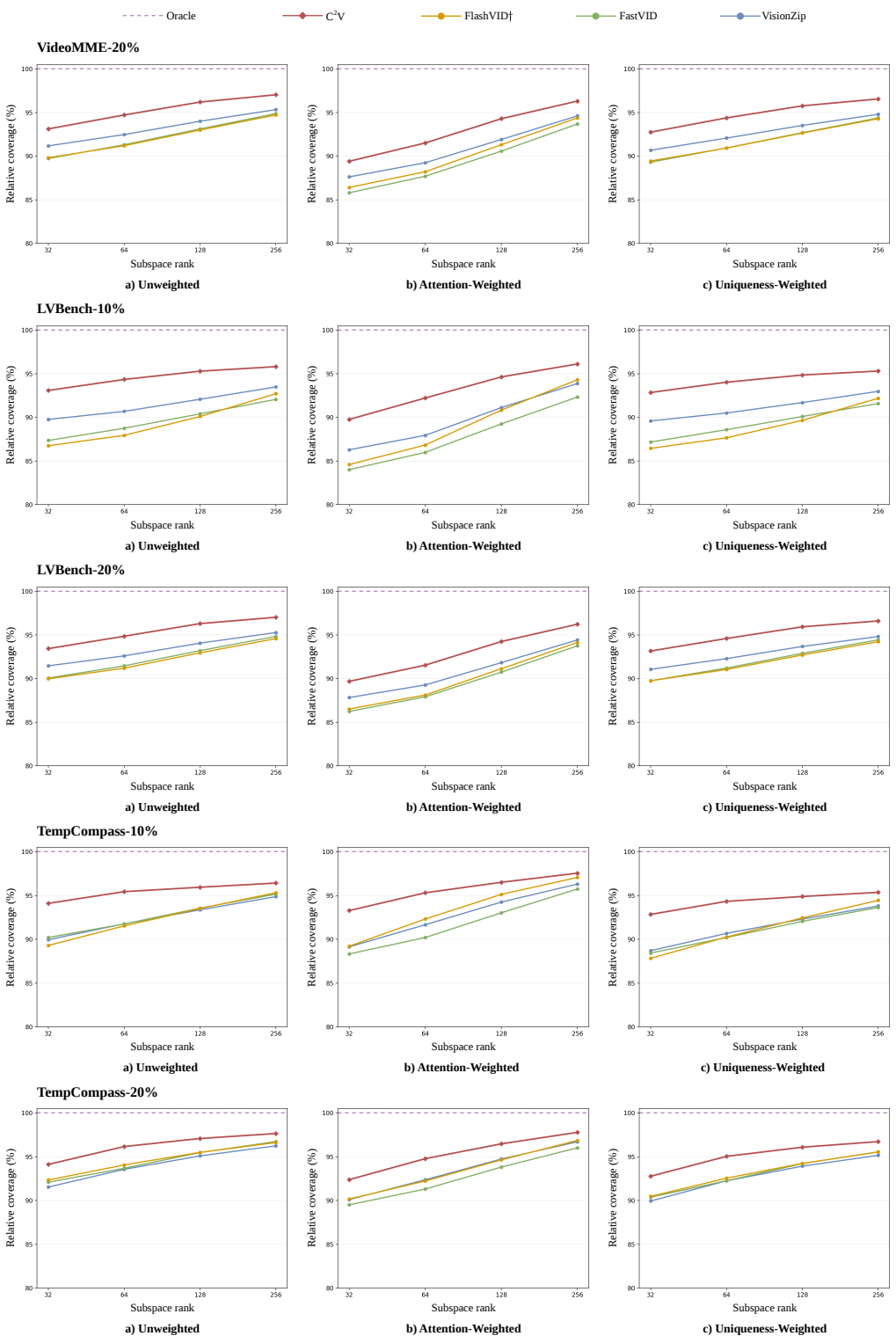

Figure D.1: Additional video-level feature coverage. Rows show VideoMME at 20% retention, LVBench at 10% and 20%, and TempCompass at 10% and 20%. Columns use unweighted, attention-weighted, and uniqueness-weighted evaluation. Dashed lines mark the same-rank optimal references (100%); higher is better.

E ADDITIONAL EFFICIENCY RESULTS

Table E.1 reports compressor time-complexity bounds and measured latency on Qwen3-VL-8B with 32 sampled frames. Latencies are averaged over 2,700 question instances from 900 VideoMME videos. Vision-encoding time includes final-layer attention extraction for VisionZip, FastVID, FlashVID†, and C^{2}V; DivPrune does not require this extraction. Compression is timed separately, excluding vision encoding and LLM computation.

Table E.1: Compressor complexity and latency on Qwen3-VL-8B. All times are in milliseconds. Vision encoding includes attention extraction when required.

Method	Time complexity	Vision encoding	Compression	
			$R = 10\%$	$R = 25\%$
DivPrune	$O(M^2D + MK_g)$	601.1	88.01	199.03
VisionZip	$O(T[NCD + N \log N])$	656.0	20.20	21.04
FastVID	$O(TN^2D + ND\sum_s P_s A_s + TN \log(TN))$	656.0	35.00	34.45
FlashVID†	$O(TN^2D + TNA^2 + TN \log(TN))$	656.0	54.40	81.44
C^{2}V	$O(TN^2D)$	656.0	25.80	35.60

Complexity analysis. The bounds are derived from the official reference implementations (Alvar et al., 2025; Yang et al., 2025; Shen et al., 2025; Fan et al., 2026), accounting for the adaptations used in our Qwen3-VL evaluation. In particular, DivPrune uses incremental minimum-distance updates, and FlashVID† uses the `attn_div` configuration. Vision encoding, attention extraction, and LLM computation are excluded from these bounds.

Here, T is the number of frames at the compressor input, N the number of tokens per frame, D the feature dimension, and $M = TN$. For VisionZip, C is the contextual-token count per frame ($C \approx 0.05N$ in our configuration); K_g is DivPrune's global output-token count. For FastVID, P_s and A_s denote the number of frames and the total contextual merging-center count in segment s, respectively. For FlashVID, A bounds the number of protected tokens per frame.

For C^{2}V, let $K = \max_t K_t$. Kernel construction and aggregation cost $O(T[N^2D+NKD+K^2D])$, while cached statistics reduce greedy selection to $O(TNK)$ (Appendix A.4). Since $K \leq N$, the overall complexity is $O(TN^2D)$. The measured latencies complement these asymptotic bounds, which do not by themselves determine practical runtime.

F Qualitative Visualizations

We complement the distribution-level analysis with visualizations of anchor selections. The overlays indicate selected source locations, not the complete information carried by the aggregated features. All examples visualize token selection on Qwen3-VL-8B at a retention ratio of 25%.

F.1 Effect of State Conditioning

Figures F.1 and F.2 compare the w/o-state MMD variant (blue) with C^{2}V (red). The selections differ around the manipulated cups and the lighting device, respectively. In both examples, C^{2}V answers correctly while the w/o-state MMD variant does not, illustrating how state conditioning can support the retention of task-relevant evidence.

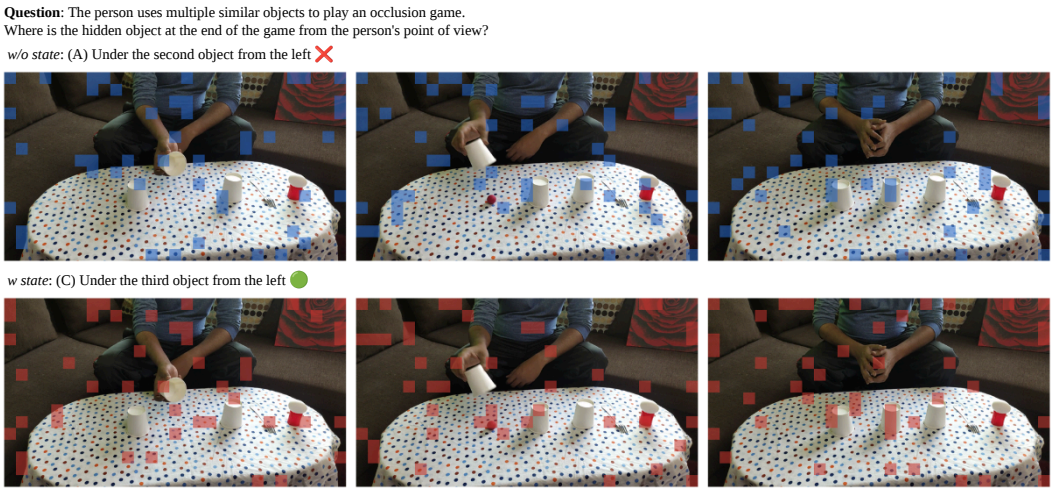

Figure F.1: Hidden-object tracking. The same three views are shown for the w/o-state MMD variant (top, blue) and C^{2}V (bottom, red). C^{2}V correctly answers (C), under the third object from the left in the person's view, whereas the w/o-state MMD variant answers (A), under the second object.

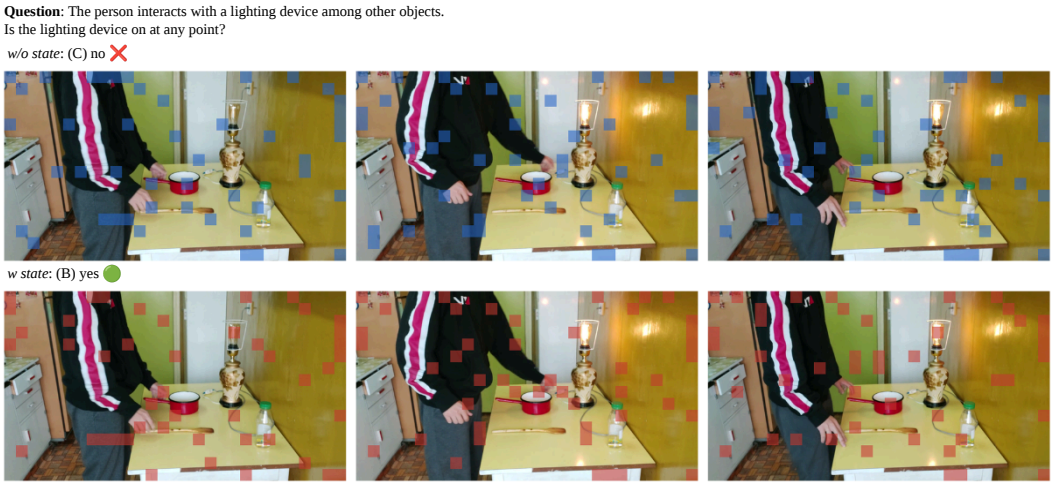

Figure F.2: Lighting-state recognition. The w/o-state MMD variant (top, blue) answers (C), no, while C^{2}V (bottom, red) correctly answers (B), yes. The device is illuminated in the middle view, where C^{2}V selects patches on the light.

F.2 Anchor Selection Across the Video

Figures F.3 and F.4 show the complete sequence of frame-wise anchor selections for papercraft and egocentric cooking videos. In each pair, source views are above and C^2V anchor overlays are below; temporal order is left to right, then top to bottom.



Figure F.3: Anchor selections in a papercraft video. Each 4×8 grid follows decorative flower construction in chronological order. Top: source views. Bottom: corresponding C^2V selections (red).



Figure F.4: Anchor selections in an egocentric cooking video. Source views (top) and C^2V selections (bottom, red) follow the video in chronological order, alternating between the grill and preparation area as hands, ingredients, and utensils change position.