V-CAST: Video Curvature-Aware Spatio-Temporal Pruning for Efficient Video Large Language Models

Xinying Lin[1,2] **Xuyang Liu**[3] **Yiyu Wang**[4] **Teng Ma**[1] **Jiasheng Li**[5] **Zichen Wen**[5] **Wenqi Ren**[1✉]

[1]Shenzhen Campus of Sun Yat-sen University [2]Shenzhen Loop Area Institute

[3]The Hong Kong Polytechnic University

[4]The Hong Kong University of Science and Technology (Guangzhou) [5]Shanghai Jiao Tong University

Homepage: https://xinyouu.github.io/V-CAST/

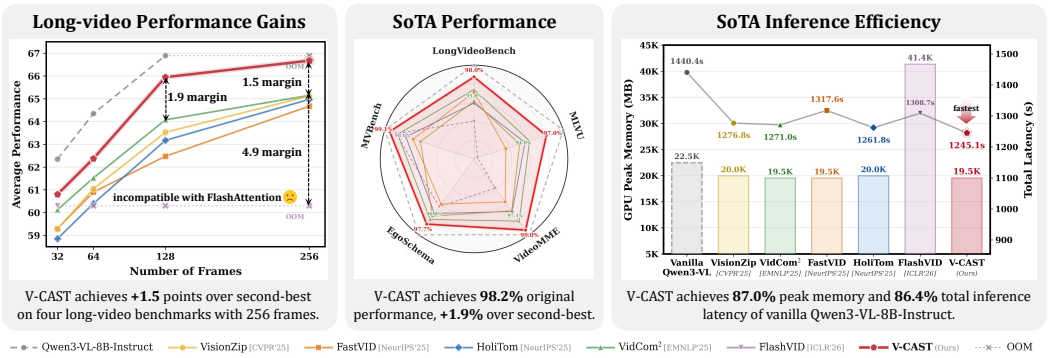


Figure 1: **Highlight of V-CAST.** Compared to existing token compression methods [46, 34, 32, 26, 11], V-CAST yields **+1.5** long-video gains, preserves **98.2%** original performance with a **+1.9%** margin, and reduces peak memory and latency to **87.0%** and **86.4%** of Qwen3-VL-8B-Instruct [2].

Abstract

Video large language models (VideoLLMs) show strong capability in video understanding, yet long-context inference is still dominated by massive redundant visual tokens in the prefill stage. We revisit token compression for VideoLLMs under a tight budget and identify a key bottleneck, namely insufficient *spatio-temporal information coverage*. Existing methods often introduce discontinuous coverage through coarse per-frame allocation or scene segmentation, and token merging can further misalign spatio-temporal coordinates under MRoPE-style discrete (t, h, w) bindings. To address these issues, we propose V-CAST (**V**ideo **C**urvature-**A**ware **S**patio-**T**emporal Pruning), a training-free, plug-and-play pruning policy for long-context video inference. V-CAST treats frame representations as a semantic trajectory and uses local curvature as a low-cost proxy for temporal turns, routing token budgets to transition regions without relying on decoder attention. It further adopts a dual-anchor spatial selection mechanism that preserves diverse and salient visual evidence, while keeping retained tokens at their original coordinates to maintain positional alignment. Extensive experiments across multiple VideoLLMs of diverse architectures and scales demonstrate that V-CAST preserves **98.2%** of the original performance with only 25% of the visual tokens, while reducing prefill latency by **51.2%** and peak memory usage by **13.0%** on Qwen3-VL-8B-Instruct.

✉ Corresponding author: renwq3@mail.sysu.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

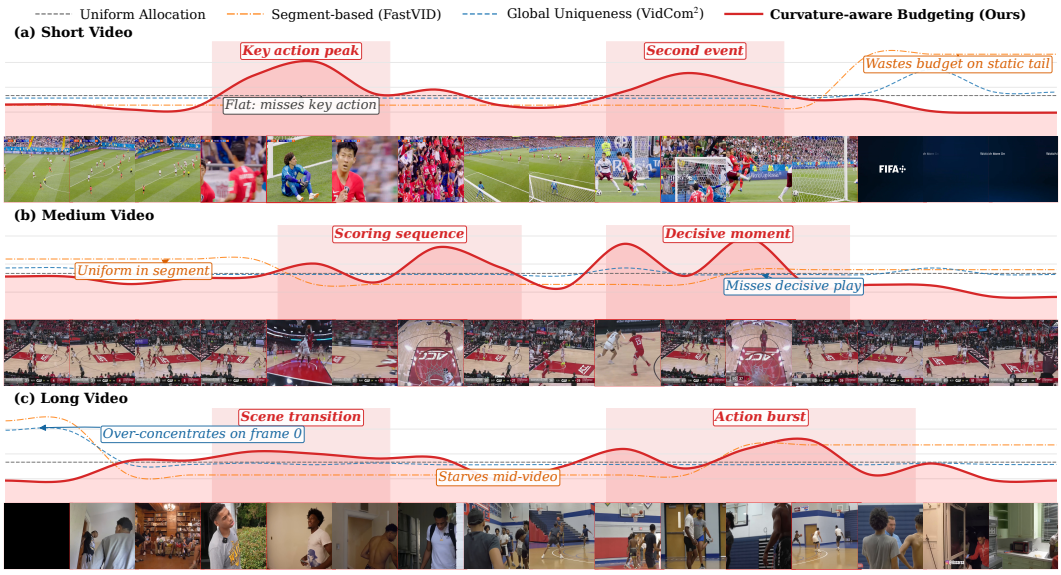


Figure 2: **Spatio-temporal budgeting profiles.** We visualize per-frame token budgets (higher means more retained tokens) on representative short, medium, and long videos. Uniform Allocation [46] stays nearly flat and misses brief evidence peaks; segment-based pipelines [34, 32] introduce boundary-sensitive budget jumps and waste tokens within redundant windows; global-uniqueness budgeting [26] can over-concentrate on a few globally distinctive frames and under-cover transitional segments. **Curvature-aware** budgeting adaptively allocates more tokens to rapid semantic changes and event boundaries, improving spatio-temporal information coverage under tight token budgets.

1 Introduction

Video large language models (VideoLLMs) [41, 6, 50, 48, 49] extend large vision-language models (LVLMs) to large-scale video data [20, 53] and have achieved strong performance on diverse video understanding tasks [12, 25, 42, 37]. However, videos introduce highly dense visual tokens from consecutive frames, so long-context prefill becomes the dominant compute and memory cost in attention-based inference. For example, advanced VideoLLM Qwen3-VL [2] supports up to a 256K context window and performs well on long-video understanding [39, 43, 54], but scaling video inputs to such long contexts becomes prohibitively expensive, which limits practical deployment [28, 33].

To alleviate this efficiency bottleneck, token compression reduces visual tokens by pruning redundant visual content [14, 27, 4, 36, 24, 18, 10, 16]. Recent works further tailor compression to videos by introducing frame-wise control, including *scene-based segmentation* pipelines [34, 32] and *frame-wise token budgeting* strategies [26, 29]. Despite these advances, under a fixed and tight token budget it remains difficult to preserve **spatio-temporal information coverage**, as reflected by the budgeting profiles in Figure 2. We find that this limitation mainly stems from ***two key factors***:

(I) Discontinuous Spatio-temporal Information: In human video perception, attention is distributed unevenly over time, yet understanding relies on continuous context because decisive moments are supported by nearby transitions. Uniform compression [46] keeps an identical budget for every frame; the flat profile in Figure 2 shows that it cannot upweight brief evidence peaks, so critical moments are easily diluted under tight token budgets. Segment-based pipelines (*e.g.*, FastVID [34] and HoliTom [32]) partition videos and compress tokens within each segment, inducing boundary-sensitive budget jumps and near-uniform allocation inside segments, weakening continuity across transitions. Global-uniqueness budgeting such as VidCom^2 [26] allocates budgets by global frame distinctiveness, which can over-concentrate on a few globally salient but not locally decisive frames, leaving transitional segments under-covered and causing discontinuous coverage under tight budgets.

(II) Misaligned Spatio-temporal Information: Advanced VideoLLMs (*e.g.*, Qwen3-VL [2]) adopt MRoPE to tie each visual token to discrete (t, h, w) coordinates for spatio-temporal reasoning. Token merging (*i.e.*, Merging) aggregates multiple visual tokens into one (*e.g.*, average pooling), which shifts the merged token off the original coordinate grid and breaks the one-to-one correspondence with MRoPE in this setting, as illustrated in Figure 3. This effect is reflected in Table 1: the

carefully designed HoliTom [32] even underperforms the naive AvgPool baseline by **2.4** points on MVBench [22], **1.8** on LongVideoBench [43], and **0.8** on VideoMME [12]. Notably, a simple pruning baseline that keeps one token per 2×2 cell already surpasses merging by a large margin, improving over HoliTom by **+2.2** on VideoMME. In contrast, token pruning selects existing tokens and therefore preserves original on-grid coordinates, maintaining positional alignment under MRoPE.

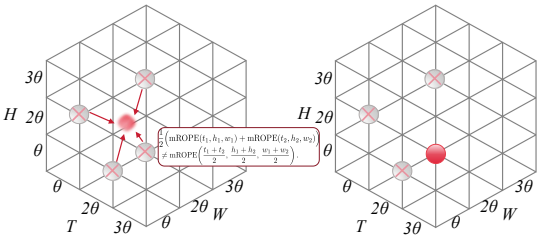
Based on the above analysis, token compression for VideoLLMs can be formulated as the following question: *how can we select a compact set of informative visual tokens under a fixed and tight token budget while preserving continuous and well-aligned spatio-temporal information coverage?*.

To answer this question, we propose **V**ideo **C**urvature-**A**ware **S**patio-**T**emporal Pruning (*i.e.*, **V-CAST**), a plug-and-play token pruning approach from the perspective of *video spatio-temporal curvature*. V-CAST couples *per-frame token budgeting* to allocate capacity to informative local temporal transitions with *content-driven token selection* to further preserve diverse spatial evidence, while naturally keeping retained tokens at their original coordinates to maintain positional alignment.

As highlighted in Figure 1, V-CAST achieves a strong accuracy–efficiency trade-off for long-context video inference, improving long-video understanding while reducing memory and latency. Specifically, it yields **+1.5** points over the second-best method on four long-video benchmarks with 256 frames, achieves **98.2%** original performance with a **+1.9%** margin, and improves efficiency to **87.0%** peak memory and **86.4%** total inference latency of vanilla Qwen3-VL-8B-Instruct.

Table 1: Comparisons of different spatial selection strategies at $R = 25\%$.

Spatial Selection	MVBench	LongVideo Bench	VideoMME			
			All	S	M	L
Qwen3-VL-8B-Instruct	68.6	60.3	64.5	76.0	60.4	57.0
(a) Token Merging						
HoliTom (NeurIPS'25)	63.0	56.8	59.7	71.4	54.6	53.1
AvgPool (2×2)	65.4	58.6	60.5	72.3	56.7	52.4
(b) Token Pruning						
Random (2×2)	66.9	57.8	61.9	73.6	58.7	53.4
First (2×2)	67.7	57.7	61.1	72.2	56.2	55.0



(a) Token Merging (b) Token Pruning

Figure 3: Token merging drifts off the discrete (t, h, w) grid and weakens MRoPE [2] bindings, while token pruning preserves on-grid coordinates.

In summary, our contributions are threefold:

1. **Spatio-temporal Coverage Rethinking.** We revisit visual token compression for VideoLLMs under tight budgets and identify *spatio-temporal information coverage* as a core bottleneck. We reveal two failure modes in prior pipelines: *discontinuous* coverage from segmentation or coarse budgeting, and *misaligned* spatio-temporal information from token merging under MRoPE.
2. **Video Curvature-Aware Spatio-Temporal Pruning.** Based on these findings, we propose **V-CAST**, a training-free, plug-and-play token pruning approach that couples *per-frame token budgeting* with *content-driven token selection* from the perspective of video spatio-temporal curvature, while keeping retained tokens at their original coordinates to preserve positional alignment.
3. **State-of-The-Art Performance-Efficiency.** Experiments on multiple benchmarks show that V-CAST achieves the best accuracy–efficiency trade-off on Qwen3-VL-8B/32B-Instruct, LLaVA-OV/Video-7B, and Qwen2.5-Omni-7B. Under 128-frame inference, V-CAST exceeds the second-best method by **+1.9** points on average, while reducing peak memory and end-to-end latency.

2 Related Work

Video Large Language Models. Large vision-language models (LVLMs) commonly follow the "ViT–Projector–LLM" paradigm, projecting visual tokens into LLM embeddings for autoregressive multimodal reasoning [23, 1, 8, 19]. VideoLLMs extend this framework with temporal modeling and video-specific training [53, 35]. LLaVA-OneVision [20] unifies image and video inputs, while LLaVA-Video [53] scales video instruction data to strengthen temporal understanding of video events. Qwen2.5-VL [3] and Keye-VL [38] adopt Multimodal Rotary Positional Embedding (MRoPE) to encode (t, h, w) positions across video frames, with Qwen3-VL [2] scaling to longer contexts and InternVL3.5 [40] leveraging video-centric training. Videos contain more tokens and higher spatio-temporal redundancy than images, making efficient real-time inference challenging [7].

Token Compression for VideoLLMs. Token compression reduces computation by pruning redundant visual tokens, often training-free and plug-and-play [5, 46]. Video-specific compression mainly follows two paradigms: **(i) Scene-based compression**, partitioning videos into temporal windows and compressing tokens within each. FastVID [34] prunes intra-window redundancy, while HoliTom [32] combines segmentation with spatio-temporal merging. These can be sensitive to boundary quality under gradual or subtle changes. **(ii) Budget-based compression**, assigning per-frame token budgets, e.g., VidCom^2 [26] uses frame uniqueness but may over-concentrate budgets and miss transitional segments, while MMG-Vid [29] combines segmentation and budgeting but still requires explicit segmentation. Overall, prior methods can produce discontinuous or misaligned spatio-temporal coverage under tight token budgets. V-CAST addresses these limitations by coordinating temporal budgeting with spatial token selection to improve coverage under aggressive compression.

3 Methodology

3.1 Preliminary

VideoLLM Pipeline. Standard VideoLLMs follow the "ViT–Projector–LLM" paradigm [53, 2]. Given a video $\mathcal{V} \in \mathbb{R}^{T\times H\times W\times 3}$, a vision encoder $\phi(\cdot)$ extracts patch features $\mathbf{Z} = \phi(\mathcal{V}) \in \mathbb{R}^{T\times P\times D}$, where P is the number of patch tokens per frame. A projector $g(\cdot)$ maps them into the LLM space as $\mathbf{X} = g(\mathbf{Z}) \in \mathbb{R}^{T\times P\times D'}$, which is flattened into $\mathbf{X}^\flat \in \mathbb{R}^{N\times D'}$ with $N = T \cdot P$. Given text tokens $\mathbf{Y} \in \mathbb{R}^{M\times D'}$, the LLM pre-fills the concatenated sequence $\mathbf{S}_0 = [\mathbf{X}^\flat; \mathbf{Y}]$ and caches layer-wise KV pairs $\{(\mathbf{K}^\ell, \mathbf{V}^\ell)\}_{\ell=1}^{L}$ for autoregressive decoding of the model's textual response.

3.2 V-CAST: Curvature-Aware Spatio-Temporal Pruning

Problem Formulation. In VideoLLMs, the visual sequence length scales as $N = T \cdot P$, which makes the pre-filling attention over $\mathbf{S}_0 = [\mathbf{X}^\flat; \mathbf{Y}]$ expensive. Token compression therefore aims to construct a shorter visual sequence $\tilde{\mathbf{X}}^\flat$ under a strict budget $B = \lfloor rN \rfloor$ ($r \in (0, 1)$), seeking to retain downstream video understanding performance. Concretely, we select an index set $\mathcal{I} \subseteq \{1, \dots, N\}$ with $|\mathcal{I}| = B$ and keep the original spatio-temporal token order, yielding $\tilde{\mathbf{X}}^\flat = \mathbf{X}^\flat[\mathcal{I}]$. Under tight budgets, the key challenge is to preserve continuous and well-aligned spatio-temporal information coverage rather than allocating tokens uniformly in time or pooling them statically in space.

Overview. Our design is guided by two questions: **(Q1)** *When should we allocate tokens across frames under a fixed budget?*. **(Q2)** *Where should we keep tokens within each frame to preserve diverse visual evidence?*. To answer them, we propose **V-CAST** (Figure 4), a training-free and plug-and-play pruning approach with two components. **(i) Curvature-Guided Temporal Allocation** (Section 3.3) rationally distributes budgets over time based on semantic transitions across frames, and **(ii) Dual-Anchor Spatial Token Selection** (Section 3.4) retains a compact yet informative set of tokens while preserving the on-grid coordinates for subsequent processing by the LLM.

3.3 Curvature-Guided Temporal Allocation

We allocate the token budget across frames by measuring how the video semantics evolves over time. Let $\mathbf{x}_{t,i} \in \mathbb{R}^{D'}$ denote the projected visual token at frame t and spatial index i. We first summarize each frame into a mean semantic vector

$$\bar{\mathbf{x}}_t = \frac{1}{P}\sum_{i=1}^{P} \mathbf{x}_{t,i} \in \mathbb{R}^{D'}, \tag{1}$$

which preserves both direction and magnitude of the frame representation. We then normalize it to a unit direction

$$\hat{\mathbf{x}}_t = \frac{\bar{\mathbf{x}}_t}{\|\bar{\mathbf{x}}_t\|_2} \in \mathbb{S}^{D'-1}, \tag{2}$$

so each frame becomes a hypersphere point encoding semantic direction. Connecting $\{\hat{\mathbf{x}}_t\}_{t=1}^{T}$ forms a representation trajectory, where sharp turns serve as a low-cost proxy for visual or semantic transitions. V-CAST uses this signal to route density toward likely evidence changes, while entropy regularization prevents over-concentration on isolated high-curvature frames.

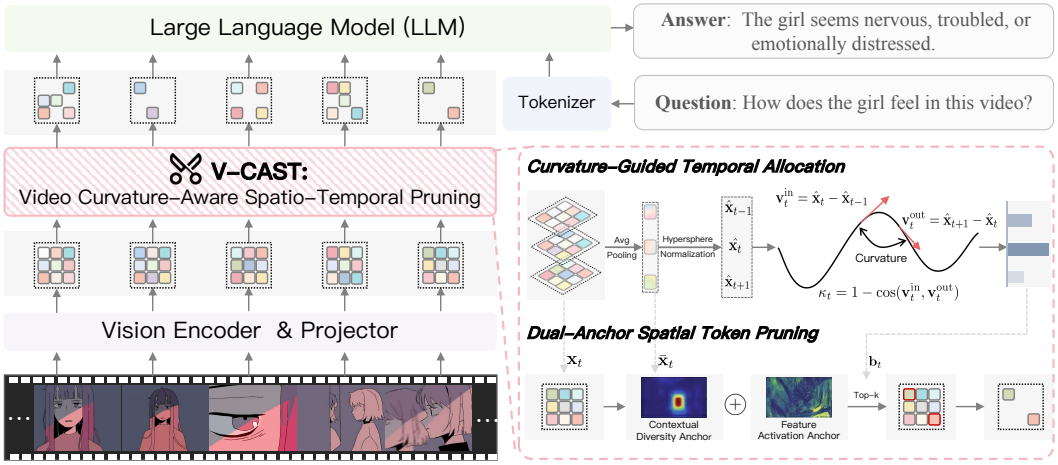


Figure 4: **Overview of V-CAST.** V-CAST formulates VideoLLM token pruning as a semantic-trajectory approximation problem under a fixed budget. It applies **Curvature-Guided Temporal Allocation** to assign per-frame budgets by tracking semantic transitions, then performs **Dual-Anchor Spatial Token Selection** to retain diverse and salient tokens within each frame.

For boundary frames, we use one-sided differences by clamping indices, *i.e.*, $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_{T+1} = \hat{\mathbf{x}}_T$. We define the incoming and outgoing directions at frame t as

$$\mathbf{v}_t^{\text{in}} = \hat{\mathbf{x}}_t - \hat{\mathbf{x}}_{t-1}, \quad \mathbf{v}_t^{\text{out}} = \hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t, \tag{3}$$

and measure the semantic curvature by the change of directions:

$$\kappa_t = 1 - \cos(\mathbf{v}_t^{\text{in}}, \mathbf{v}_t^{\text{out}}), \tag{4}$$

where $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2 + \epsilon}$ and ϵ is a small constant.

Given the global budget B, we allocate a budget density p_t over frames by solving an entropy-regularized allocation problem:

$$\max_{\mathbf{p} \in \Delta^{T-1}} \sum_{t=1}^{T} p_t \kappa_t + \tau \mathcal{H}(\mathbf{p}), \quad \mathcal{H}(\mathbf{p}) = -\sum_{t=1}^{T} p_t \log p_t, \tag{5}$$

where Δ^{T-1} is the probability simplex and τ is a temperature hyperparameter controlling allocation smoothness, and we use a fixed default $\tau = 0.7$. This objective yields a closed-form solution

$$p_t^* = \frac{1}{\mathcal{Z}} \exp\left(\frac{\kappa_t}{\tau}\right), \quad \mathcal{Z} = \sum_{j=1}^{T} \exp\left(\frac{\kappa_j}{\tau}\right). \tag{6}$$

We convert $\mathbf{p}^*$ into integer budgets $\{b_t\}_{t=1}^T$ while enforcing $\sum_t b_t = B$ and $k_{\min} \le b_t \le P$, redistributing tokens as needed to satisfy these constraints, so high-curvature transitions receive more tokens without overly concentrating the budget.

3.4 Dual-Anchor Spatial Token Selection

Given the per-frame budget $b_t \ll P$, we select spatial tokens that preserve both diversity and saliency within each frame. We compute a **Dual-Anchor** score for each token $\mathbf{x}_{t,i}$:

$$s_{t,i} = \underbrace{\left(1 - \cos(\mathbf{x}_{t,i}, \bar{\mathbf{x}}_t)\right)}_{\text{Contextual Diversity Anchor}} + \underbrace{\mathcal{N}\left(\|\mathbf{x}_{t,i}\|_2\right)}_{\text{Feature Activation Anchor}}, \tag{7}$$

where $\mathcal{N}(\cdot)$ denotes min–max normalization.

Contextual Diversity Anchor. The first term favors tokens that deviate from the global frame context $\bar{\mathbf{x}}_t$, which often correspond to subjects, interactions, or locally distinctive regions, and helps avoid

keeping many redundant background patches. **Feature Activation Anchor.** The second term uses the ℓ_2-norm as a lightweight saliency cue; empirically, higher-norm ViT features tend to indicate stronger activations and visually informative patterns.

We form $\mathcal{S}_t$ by selecting the Top-b_t tokens with the largest $s_{t,i}$ and keep their original spatial coordinates, which preserves on-grid positional alignment under MRoPE. Computing $\{s_{t,i}\}$ is linear in P, and Top-b_t selection can be implemented efficiently with partial selection in practice.

3.5 Theoretical Insight on Spatio-temporal Coverage

Temporal Allocation via Curvature. V-CAST represents frame semantics as $\{\hat{\mathbf{x}}_t\}_{t=1}^{T} \subset \mathbb{S}^{D'-1}$ and allocates budgets based on direction changes. Let $\Delta_t = \hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t$ and $\kappa_t = 1 - \cos(\Delta_{t-1}, \Delta_t)$. Since $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2 + \epsilon}$, larger κ_t indicates a larger turn between adjacent directions, *i.e.*, a more significant semantic transition. We route the global budget by

$$\max_{\mathbf{p} \in \Delta^{T-1}} \sum_{t=1}^{T} p_t \kappa_t + \tau \mathcal{H}(\mathbf{p}) \quad \Rightarrow \quad p_t^* = \frac{1}{\mathcal{Z}} \exp\Big(\frac{\kappa_t}{\tau}\Big), \tag{8}$$

so frames with larger κ_t receive exponentially more budget while $\tau > 0$ prevents collapse to a few frames, which improves temporal coverage under tight budgets.

Spatial Selection via Diversity–Saliency. For each token, V-CAST uses

$$s_{t,i} = \big(1 - \cos(\mathbf{x}_{t,i}, \bar{\mathbf{x}}_t)\big) + \mathcal{N}(\|\mathbf{x}_{t,i}\|_2). \tag{9}$$

Let $\tilde{\mathbf{x}}_{t,i} = \mathbf{x}_{t,i} / \|\mathbf{x}_{t,i}\|_2$ and $\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_t / \|\bar{\mathbf{x}}_t\|_2$. Then

$$1 - \cos(\mathbf{x}_{t,i}, \bar{\mathbf{x}}_t) = 1 - \cos(\tilde{\mathbf{x}}_{t,i}, \hat{\mathbf{x}}_t) = \tfrac{1}{2} \|\tilde{\mathbf{x}}_{t,i} - \hat{\mathbf{x}}_t\|_2^2, \tag{10}$$

which favors directionally diverse tokens around $\hat{\mathbf{x}}_t$, while $\mathcal{N}(\|\mathbf{x}_{t,i}\|_2)$ favors salient activations. Selecting Top-b_t by $s_{t,i}$ thus preserves diverse yet informative spatial evidence, and pruning keeps all retained tokens on-grid, maintaining positional alignment under MRoPE.

4 Experiments

4.1 Experimental Setting

Benchmarks. We evaluate on several multimodal video understanding benchmarks covering videos of diverse durations and scenarios, including MVBench [22], LongVideoBench [43], MLVU [54], VideoMME [12], and EgoSchema [30]. We adopt the LMMs-Eval framework [51] for evaluation. Detailed information on benchmark characteristics and tasks is provided in Appendix A.

Implementations. We validate token compression on various state-of-the-art VideoLLMs, including Qwen3-VL [2], LLaVA-Video [53], LLaVA-OneVision [20], and Qwen2.5-Omni [45]. We follow each backbone's official frame setting for LLaVA series models, and further test longer contexts with Qwen3-VL from 32 to 256 frames and Qwen2.5-Omni at 768 audio-visual frames. All experiments are conducted on NVIDIA H100-80GB HBM3 GPUs. Appendix B provides more details.

Baselines. We compare our V-CAST against representative token compression methods for VideoLLMs, including VisionZip [46], FastVID [34], HoliTom [32], VidCom^2 [26] and FlashVID [11]. We re-implement all baseline token compression methods under the same experimental settings.

4.2 Main Comparisons

Comparisons on Qwen3-VL-8B-Instruct. Table 2 compares V-CAST with existing methods on Qwen3-VL-8B-Instruct and highlights three key observations. **(i) On MRoPE-based Qwen3-VL, pruning tends to outperform merging:** Pruning methods, such as VidCom^2 [26] and V-CAST, consistently outperform token merging methods, such as HoliTom [32] and FastVID [34]. This trend is expected because Qwen3-VL adopts a three-dimensional MRoPE for video, and merging shifts token positions away from their original spatio-temporal coordinates, which weakens spatio-temporal alignment during downstream video understanding. **(ii) Stronger gains with longer inputs:** V-CAST

Table 2: **Performance comparison with other baselines on Qwen3-VL-8B-Instruct across benchmarks under multiple settings.** "Average" shows the mean performance across benchmarks.

Methods	MVBench	LongVideo Bench	MLVU	VideoMME				Average	
				Overall	Short	Medium	Long	Score	%
Max Input Frames=64									
Qwen3-VL-8B-Instruct	69.2	62.8	68.9	66.9	78.8	66.2	55.8	67.0	100.0
Retention Ratio=25%									
VisionZip (CVPR'25)	64.8	58.9	63.4	63.1	73.7	60.3	55.4	62.6	93.4
VidCom^2 (EMNLP'25)	67.5	59.6	64.0	64.9	75.4	**63.4**	**55.9**	64.0	95.5
FastVID (NeurIPS'25)	66.8	60.4	65.0	62.3	73.9	60.7	52.2	63.6	94.9
HoliTom (NeurIPS'25)	64.8	58.9	63.4	62.7	74.4	60.6	53.2	62.5	93.3
FlashVID (ICLR'26)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-	-
V-CAST (Ours)	**68.4**	**61.2**	**65.4**	**65.1**	**76.8**	63.2	55.3	**65.0**	**97.0**
Retention Ratio=15%									
VisionZip (CVPR'25)	62.0	57.7	61.4	60.4	70.3	58.4	52.6	60.4	90.1
VidCom^2 (EMNLP'25)	64.3	57.4	60.3	62.9	72.6	61.3	**54.7**	61.2	91.3
FastVID (NeurIPS'25)	64.9	**59.7**	63.3	60.5	72.2	57.2	52.1	62.1	92.7
HoliTom (NeurIPS'25)	62.7	58.3	61.8	60.9	72.3	58.2	52.2	60.9	90.9
FlashVID (ICLR'26)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-	-
V-CAST (Ours)	**66.3**	59.6	**64.4**	**63.9**	**76.3**	**61.4**	53.8	**63.6**	**94.9**
Max Input Frames=32									
Qwen3-VL-8B-Instruct	68.6	60.3	63.5	64.5	76.0	60.4	57.0	64.2	100.0
Retention Ratio=25%									
VisionZip (CVPR'25)	62.2	56.7	60.8	60.1	69.6	56.7	54.2	60.0	93.5
VidCom^2 (EMNLP'25)	67.0	58.0	60.6	62.4	72.1	59.1	**56.1**	62.0	96.6
FastVID (NeurIPS'25)	67.3	58.7	60.7	60.5	72.0	56.8	52.7	61.8	96.3
HoliTom (NeurIPS'25)	63.0	56.8	61.2	59.7	71.4	54.6	53.1	60.2	93.8
FlashVID (ICLR'26)	67.5	**58.8**	61.7	62.3	**74.4**	58.7	53.9	62.6	97.5
V-CAST (Ours)	**67.9**	58.2	**62.7**	**63.5**	**74.4**	**60.2**	55.8	**63.1**	**98.2**
Retention Ratio=15%									
VisionZip (CVPR'25)	60.1	56.2	60.0	58.2	66.9	54.9	52.9	58.6	91.3
VidCom^2 (EMNLP'25)	64.2	56.0	57.7	59.6	68.9	56.7	53.3	59.4	92.5
FastVID (NeurIPS'25)	66.1	57.1	59.0	58.3	69.6	54.3	51.1	60.1	93.6
HoliTom (NeurIPS'25)	60.0	55.8	59.6	58.3	68.9	54.7	51.7	58.4	91.0
FlashVID (ICLR'26)	**66.5**	**57.8**	60.0	61.4	**72.4**	57.6	54.1	61.4	95.6
V-CAST (Ours)	66.2	57.7	**60.1**	**62.0**	71.9	**58.3**	**55.9**	**61.5**	**95.8**

performs better under the 64-frame setting than under the 32-frame setting at the same retention ratio, since it removes temporally redundant tokens while preserving salient motion and event cues, thereby retaining rich spatio-temporal information for long-context reasoning. **(iii) Extreme compression robustness:** V-CAST remains robust under aggressive compression. For Max Input Frames$=32$, it achieves **63.1** at $R = 25\%$ (preserving **98.2%** of the uncompressed performance) and maintains **61.5** at $R = 15\%$ (corresponding to **95.8%** retention), while ranking first by average benchmark score in both settings. Notably, FlashVID runs out of memory under 64-frame inputs, likely because its attention-guided compression is executed inside the LLM and incurs higher activation and attention overhead, which also hinders efficient FlashAttention [9] execution in long-video settings.

Comparisons on LLaVA-OV-7B and LLaVA-Video-7B. Table 3 compares V-CAST with a broader set of baselines [44, 52, 17, 13] on the LLaVA series VideoLLMs, showing consistent improvements without training. These backbones use standard RoPE rather than Qwen3-VL's 3D MRoPE, indicating broad compatibility with common positional encodings. At $R = 25\%$, V-CAST achieves an average score of **58.8** on LLaVA-OV-7B and **59.9** on LLaVA-Video-7B, preserving **99.5%** and **95.4%** of original performance. The results indicate that the same budgeting and token selection rules reliably transfer across VideoLLM backbones and generalize to videos of varying lengths.

Comparisons on Larger VideoLLM Qwen3-VL-32B-Instruct. Table 4 compares V-CAST with existing methods on Qwen3-VL-32B-Instruct, showing that the token selection policy transfers to larger dense VideoLLMs. At $R = 25\%$, V-CAST preserves **98.1%** of original performance, achieving the best average score among compressed baselines and outperforming the second-best VidCom^2 [26] by **1.3%**. Unlike attention-guided methods, V-CAST is prefill-side, attention-free, and on-grid, easily plug-and-play with larger VideoLLMs without additional training or changes to model weights.

Table 3: **Performance comparison on LLaVA series models across different benchmarks.** We use the default input settings of 32 frames for LLaVA-OV-7B and 64 frames for LLaVA-Video-7B.

Methods	MVBench	LongVideoBench	MLVU	VideoMME				Average	
				Overall	Short	Medium	Long	Score	%
LLaVA-OV-7B	58.3	56.6	63.1	58.4	69.9	56.7	48.8	59.1	100.0
Retention Ratio=30%									
DyCoke (CVPR'25)	56.6	54.7	60.3	56.1	67.1	54.6	46.6	56.9	96.3
Retention Ratio=25%									
FastV (ECCV'24)	55.5	53.3	59.6	55.3	65.0	53.8	47.0	55.9	94.6
PDrop (CVPR'25)	55.3	51.3	57.1	55.5	64.7	53.1	48.7	54.8	92.7
SparseVLM (ICML'25)	56.4	53.9	60.7	57.3	68.4	55.2	48.1	57.1	96.6
VisionZip (CVPR'25)	56.9	56.0	62.9	58.0	68.9	57.4	47.6	58.5	99.0
PruneVid (ACL'25)	55.7	55.1	**63.4**	57.0	68.8	54.4	47.7	57.8	97.8
FrameFusion (ICCV'25)	56.0	54.8	61.7	57.5	68.2	55.7	48.6	57.5	97.3
FastVID (NeurIPS'25)	56.5	56.3	60.9	58.3	69.4	**58.2**	47.2	58.0	98.1
VidCom^2 (EMNLP'25)	57.0	55.4	62.8	58.4	69.3	56.3	**49.4**	58.4	98.8
V-CAST (Ours)	**57.4**	**56.4**	62.9	**58.6**	**70.7**	56.0	49.1	**58.8**	**99.5**
LLaVA-Video-7B	60.4	58.9	67.3	64.4	77.3	62.4	53.4	62.8	100.0
Retention Ratio=30%									
DyCoke (CVPR'25)	57.5	55.5	60.6	61.3	73.4	59.3	51.2	58.7	93.5
Retention Ratio=25%									
FastV (ECCV'24)	52.1	54.8	57.8	58.6	68.7	58.4	48.7	55.8	88.9
SparseVLM (ICML'25)	55.4	54.2	58.9	60.1	71.1	59.1	50.1	57.2	91.1
VisionZip (CVPR'25)	57.9	56.3	**62.6**	62.5	73.6	**62.3**	51.9	59.8	95.2
HoliTom (NeurIPS'25)	**58.4**	57.1	60.5	**63.0**	**74.6**	**62.3**	**52.1**	59.8	95.2
VidCom^2 (EMNLP'25)	57.0	57.1	58.7	61.7	73.0	61.7	50.0	58.6	93.4
V-CAST (Ours)	58.0	**57.3**	62.4	61.9	73.9	60.3	51.3	**59.9**	**95.4**

Table 4: **Performance comparison with other baselines on the larger model Qwen3-VL-32B-Instruct across different benchmarks.** We use the 32-frame input setting.

Methods	MVBench	LongVideoBench	MLVU	VideoMME				Average	
				Overall	Short	Medium	Long	Score	%
Qwen3-VL-32B-Instruct	73.2	62.4	66.1	69.3	78.3	67.1	62.4	67.8	100.0
Retention Ratio=25%									
VisionZip (CVPR'25)	62.9	60.0	62.2	64.8	72.8	63.1	58.6	62.5	92.2
VidCom^2 (EMNLP'25)	70.2	60.4	**64.9**	67.0	75.6	64.3	**61.2**	65.6	96.8
FastVID (NeurIPS'25)	71.0	60.9	64.5	65.1	75.1	62.6	57.7	65.4	96.5
HoliTom (NeurIPS'25)	62.3	60.2	63.2	64.9	74.2	61.2	59.1	62.6	92.3
FlashVID (ICLR'26)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-	-
V-CAST (Ours)	**71.8**	**61.5**	64.7	**68.1**	**77.7**	**65.9**	60.8	**66.5**	**98.1**

Comparisons on MoE-based VideoLLM Qwen3-VL-30B-A3B-Instruct. Table 5 compares V-CAST with baseline methods on Qwen3-VL-30B-A3B-Instruct and demonstrates its effectiveness under large-scale settings. Notably, the vanilla model and FlashVID [11] encounter OOM errors under 64-frame inputs, while V-CAST enables stable and efficient inference. Under $R = 25\%$, V-CAST achieves the best average score of **68.1** across all benchmarks. V-CAST shows clear advantages on long-video benchmarks, achieving the best results on LongVideoBench [43] with **66.6** and on VideoMME (Long) [12] with **59.8**. These results indicate that V-CAST scales well to MoE-based VideoLLMs and better preserves long-range spatio-temporal information under compression.

Scaling with More Input Frames. Figure 5 evaluates performance as the maximum number of input frames increases from 32 to 256 on LongVideoBench [43], MLVU [54], EgoSchema [30], and VideoMME-Long [12]. More frames generally improve the vanilla model and most compressed variants, suggesting that longer temporal contexts provide useful evidence. However, FlashVID [11] runs out of memory at 64 frames and above under our protocol, whereas V-CAST remains feasible without attention-guided compression. At 256 frames, V-CAST outperforms the second-best VidCom^2 [26] by **+1.5** points on average and widens the margin over segment-based methods [32, 34] and uniform compression [46]. These results show that curvature-guided routing better preserves temporal evidence coverage across extended video sequences as the frame budget grows.

Comparisons on Inference Efficiency. Table 6 compares inference efficiency on Qwen3-VL-8B-Instruct. V-CAST achieves the best overall trade-off, reducing prefill latency by **51.2%** and peak memory by **13.0%** while maintaining strong throughput. By contrast, FlashVID [11] increases peak

Table 5: **Performance comparison with other baselines on the MoE-based model Qwen3-VL-30B-A3B-Instruct across different benchmarks.** We use the 64-frame input setting.

Methods	MVBench	LongVideo Bench	MLVU	VideoMME Overall	VideoMME Short	VideoMME Medium	VideoMME Long	Average Score
Qwen3-VL-30B-A3B-Instruct	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-
Retention Ratio=25%								
VisionZip (CVPR'25)	61.3	62.4	64.1	62.7	70.7	61.3	56.0	62.6
VidCom^2 (EMNLP'25)	**69.7**	65.5	**68.5**	68.1	**79.1**	**68.0**	58.9	68.0
FastVID (NeurIPS'25)	68.2	60.7	64.8	62.4	70.3	61.6	55.2	64.0
HoliTom (NeurIPS'25)	66.0	63.9	64.7	64.6	73.6	63.9	56.2	64.8
FlashVID (ICLR'26)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-
V-CAST (Ours)	69.2	**66.6**	68.2	**68.2**	78.9	65.9	**59.8**	**68.1**

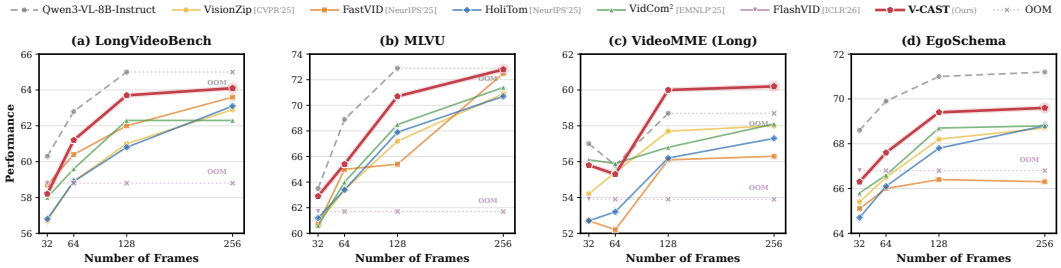


Figure 5: **Performance comparisons with more frames (32 → 256).** V-CAST improves accuracy and scales to longer inputs, while some baselines show limited gains or OOM at larger frame counts.

Table 6: **Efficiency comparison on Qwen3-VL-8B-Instruct on VideoMME at $R = 25\%$ (max input frames = 32).** "Prefill Latency": prompt-to-first-token; "LLM Generation Latency": time for LLM-only response generation; "Total Latency": total time to complete VideoMME evaluation; "Throughput": number of VideoMME samples processed per second.

Methods	Prefill ↓ Latency (s)	LLM Generation ↓ Latency (s)	Total Latency ↓ (s)	GPU Peak ↓ Memory (MB)	Throughput ↑ (item/s)	Performance ↑
Qwen3-VL-8B-Instruct	243.6	280.9	1440.4	22478.0	1.87	64.5
Random	116.9 (↓52.0%)	147.3 (↓47.6%)	1271.9 (↓11.7%)	19547.5 (↓13.0%)	2.12 (1.13×)	61.5 (↓3.0)
VisionZip (CVPR'25)	155.6 (↓36.1%)	190.5 (↓32.2%)	1276.8 (↓11.4%)	19974.7 (↓11.2%)	2.11 (1.13×)	60.1 (↓4.4)
VidCom^2 (EMNLP'25)	119.0 (↓51.1%)	150.2 (↓46.5%)	1271.0 (↓11.8%)	**19547.5** (↓13.0%)	2.12 (1.13×)	62.4 (↓2.1)
FastVID (NeurIPS'25)	120.6 (↓50.5%)	153.6 (↓45.3%)	1317.6 (↓8.5%)	**19547.5** (↓13.0%)	2.05 (1.10×)	60.5 (↓4.0)
HoliTom (NeurIPS'25)	134.3 (↓44.9%)	160.8 (↓42.8%)	1261.8 (↓12.4%)	19974.7 (↓11.2%)	2.14 (1.14×)	59.7 (↓4.8)
FlashVID (ICLR'26)	146.6 (↓39.8%)	175.0 (↓37.7%)	1308.7 (↓9.1%)	41363.6 (↑84.0%)	2.06 (1.10×)	62.3 (↓2.2)
V-CAST (Ours)	**118.8** (↓51.2%)	**149.9** (↓46.7%)	**1245.1** (↓13.6%)	**19547.5** (↓13.0%)	**2.17** (1.16×)	**63.5** (↓1.0)

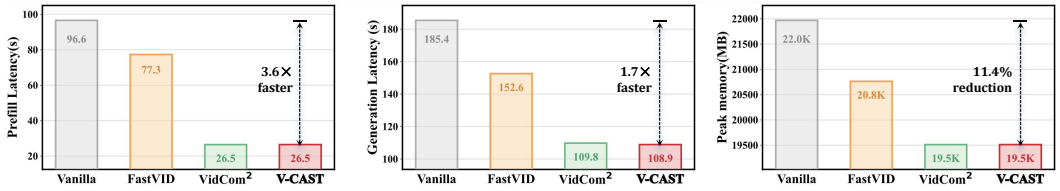


Figure 6: **Efficiency comparison on LLaVA-OneVision-7B.** We compare Vanilla, FastVID, VidCom^2, and V-CAST on Prefill Latency, Generation Latency, and Peak Memory.

memory by **84.0%** over the uncompressed model, making it more prone to OOM in long-context inference. Figure 6 shows similar gains on LLaVA-OneVision-7B, where V-CAST achieves **3.6**× faster prefill, **1.7**× faster generation, and **11.4%** lower peak memory. These results show that V-CAST delivers practical inference acceleration across VideoLLMs with different model architectures.

4.3 Ablation Study and Analysis

Ablation on Temporal Allocation. Table 7 evaluates temporal allocation under $R = 25\%$. Uniform Allocation assigns the same budget to every frame, leading to weaker results because it cannot emphasize brief evidence moments. Global Uniqueness [26] improves over Uniform by allocating more tokens to globally distinctive frames, but can over-concentrate budget and miss transitional evidence. In contrast, our Curvature-Aware allocation routes budget to local semantic turns and

Table 7: **Ablation on Temporal Allocation.** Evaluating temporal routing metrics on Qwen3-VL-8B-Instruct at $R = 25\%$ under different maximum input frames.

(a) Max Input Frames= 32.

Temporal Allocation	MLVU	VideoMME			Average
		Overall	Short	Medium	Score
Qwen3-VL-8B-Instruct	63.5	64.5	76.0	60.4	64.0
Uniform Allocation	61.6	62.9	73.9	60.0	62.3
Random Allocation	62.0	62.6	73.7	59.9	62.3
Global Uniqueness	62.1	63.0	74.1	59.9	62.6
Curvature-Aware (Ours)	**62.7**	**63.5**	**74.4**	**60.2**	**63.1**

(b) Max Input Frames= 64.

Temporal Allocation	MLVU	VideoMME			Average
		Overall	Short	Medium	Score
Qwen3-VL-8B-Instruct	68.9	66.9	78.8	66.2	67.9
Uniform Allocation	65.2	64.7	76.6	62.7	65.0
Random Allocation	64.7	64.4	75.9	62.9	64.6
Global Uniqueness	65.1	64.8	76.2	62.9	65.0
Curvature-Aware (Ours)	**65.4**	**65.1**	**76.8**	**63.2**	**65.3**

achieves the optimal performance. Appendix Figure 7 further visualizes per-frame budgets, showing that V-CAST produces targeted peaks around transition intervals.

Ablation on Spatial Selection. Table 8 compares spatial token selection variants under the same settings. Contextual Diversity Only encourages diversity but may suppress important, context-consistent evidence, while Feature Activation Only captures salient responses but can keep redundant high-activation patches and miss complementary cues. Dual-Anchor combines both signals and achieves the best overall performance by preserving diverse and informative evidence within each frame.

Table 8: **Ablation on Spatial Selection.** Evaluating spatial selection anchors on Qwen3-VL-8B at $R = 25\%$.

Spatial Selection	MVBench	MLVU	VideoMME			Average
			Overall	Short	Medium	Score
Qwen3-VL-8B-Instruct	68.6	63.5	64.5	76.0	60.4	65.5
Contextual Diversity Only	67.5	61.6	61.2	71.3	57.1	63.4
Feature Activation Only	63.0	58.6	61.4	70.0	57.2	61.0
Dual-Anchor (Ours)	**67.9**	**62.7**	**63.5**	**74.4**	**60.2**	**64.7**

5 Conclusion

In this work, we revisit token compression for VideoLLMs under tight budgets and identify two critical issues: discontinuous spatio-temporal information coverage caused by coarse allocation or segmentation, and spatio-temporal misalignment introduced by token merging under MRoPE. Based on these insights, we propose V-CAST, a training-free and plug-and-play token pruning approach that allocates temporal budgets according to local changes in the frame-representation trajectory. V-CAST combines curvature-guided temporal allocation with dual-anchor spatial selection to preserve continuous and well-aligned spatio-temporal evidence while keeping retained tokens at their original coordinates. Extensive experiments across multiple VideoLLMs and benchmarks confirm that V-CAST achieves state-of-the-art performance and efficiency for video understanding.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. U24B20175 and 62322216), the Scientific Research Innovation Capability Support Project for Young Faculty (Grant No. SRICSPYF-ZY2025012), the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (Grant No. GML-KF-26-02), Guangdong S&T Program (Grant No. 2026B0101090001), and the Shenzhen Science and Technology Program (Grant No. SYSRD20250529113401002).

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[2] 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. *arXiv preprint arXiv:2511.21631*, 2025.

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

[4] Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3489–3499, 2026.

[5] 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*, 2024.

[6] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *Proceedings of the International Conference on Learning Representations*, pages 18227–18246, 2025.

[7] Yuzhe Chen, Jiale Cao, Xuyang Liu, Jin Xie, Aiping Yang, and Yanwei Pang. Stdec: Spatio-temporal stability guided decoding for dllms. *arXiv preprint arXiv:2604.06330*, 2026.

[8] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

[9] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Proceedings of the International Conference on Learning Representations*, pages 35549–35562, 2024.

[10] Yue Ding, Yiyan Ji, Jungang Li, Xuyang Liu, Xinlong Chen, Junfei Wu, Bozhou Li, Bohan Zeng, Yang Shi, Yushuo Guan, Yuanxing Zhang, Jiaheng Liu, Qiang Liu, Pengfei Wan, and Liang Wang. OmniSIFT: Modality-asymmetric token compression for efficient omni-modal large language models. In *Proceedings of the International Conference on Machine Learning*, volume 306, pages 24925–24941, 2026.

[11] 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 *The Fourteenth International Conference on Learning Representations*, 2026.

[12] 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*, pages 24108–24118, 2025.

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

[14] Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Honggang Chen, Donglin Wang, Qingsen Yan, and Siteng Huang. Filter, correlate, compress: Training-free token reduction for mllm acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4601–4609, 2026.

[15] Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. Worldsense: Evaluating real-world omnimodal understanding for multimodal llms. In *International Conference on Learning Representations*, volume 2026, pages 52423–52443, 2026.

[16] Haoyang Huang, Wenjie Huang, Tianqi Xu, Hongyaoxing Gu, Kang Tan, Yikai Fu, Yuhao Shen, Tianyu Liu, Baolin Zhang, Jun Zhang, et al. Omnidelta: Skill-driven budget allocation for token compression in omnillms. *arXiv preprint arXiv:2607.25669*, 2026.

[17] 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*, pages 19959–19973, 2025.

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

[19] Jing Jin, Hao Liu, Yan Bai, Yihang Lou, Zhenke Wang, Tianrun Yuan, Juntong Chen, Yongkang Zhu, Fanhu Zeng, Xuanyu Zhu, Tao Feng, and Yige Xu. Unveiling fine-grained visual traces: Evaluating multimodal interleaved reasoning chains in multimodal stem tasks. *arXiv preprint arXiv:2604.19697*, 2026.

[20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[21] Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Jiafu Tang, Zhenghao Song, Dingling Zhang, et al. Omnivideobench: Towards audio-visual understanding evaluation for omni mllms. In *Proceedings of the International Conference on Learning Representations*, 2026.

[22] 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*, pages 22195–22206, 2024.

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26286–26296, 2024.

[24] Xuyang Liu, Xiyan Gui, Yuchao Zhang, and Linfeng Zhang. Mixing importance with diversity: Joint optimization for kv cache compression in large vision-language models. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pages 125061–125081, 2026.

[25] Xuyang Liu, Shihao Wang, Xinyu Wei, Minghan Li, Zeqing Wang, Zuhao Yang, Zichen Wen, and Lei Zhang. Omnicompass: A holistic benchmark for omni-modal video grounding and understanding. *arXiv preprint arXiv:2610.18829*, 2026.

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

[27] Xuyang Liu, Ziming Wang, Junjie Chen, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Siteng Huang, and Honggang Chen. Global compression commander: Plug-and-play inference acceleration for high-resolution large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 7350–7358, 2026.

[28] Xuyang Liu, Zichen Wen, Shaobo Wang, Junjie Chen, Zhishan Tao, Yubo Wang, Xiangqi Jin, Chang Zou, Yiyu Wang, Chenfei Liao, et al. Shifting ai efficiency from model-centric to data-centric compression. *arXiv preprint arXiv:2505.19147*, 2025.

[29] 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, pages 24253–24261, 2026.

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

[31] Le Thien Phuc Nguyen, Zhuoran Yu, Samuel Low Yu Hang, Subin An, Jeongik Lee, Yohan Ban, SeungEun Chung, Thanh-Huy Nguyen, JuWan Maeng, Soochahn Lee, and Yong Jae Lee. See, hear, and understand: Benchmarking audiovisual human speech understanding in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pages 2272–2283, June 2026.

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

[33] 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*, 2025.

[34] 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 *Proceedings of the Advances in Neural Information Processing Systems*, volume 38, pages 123553–123581, 2025.

[35] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-XL: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26160–26169, 2025.

[36] 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*, pages 18992–19001, 2025.

[37] Keda Tao, Yuhua Zheng, Jia Xu, Wenjie Du, Kele Shao, Hesong Wang, Xueyi Chen, Xin Jin, Junhan Zhu, Bohan Yu, Weiqiang Wang, Jian Liu, Can Qin, Yulun Zhang, Ming-Hsuan Yang, and Huan Wang. LVOmniBench: Pioneering long audio-video understanding evaluation for omnimodal llms. *arXiv preprint arXiv:2603.19217*, 2026.

[38] Kwai Keye Team, Biao Yang, Bin Wen, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl technical report. *arXiv preprint arXiv:2507.01949*, 2025.

[39] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025.

[40] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[41] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[42] Yiyu Wang, Xuyang Liu, Xiyan Gui, Xinying Lin, Boxue Yang, Chenfei Liao, Tailai Chen, and Linfeng Zhang. Accelerating streaming video large language models via hierarchical token compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18523–18533, 2026.

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

[44] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[45] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

[46] 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*, pages 19792–19802, 2025.

[47] Yudong Yang, Jimin Zhuang, Guangzhi Sun, Changli Tang, Yixuan Li, Peihan Li, Yifan Jiang, Wei Li, Zejun Ma, and Chao Zhang. Audio-centric video understanding benchmark without text shortcut. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025.

[48] Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. LongVT: Incentivizing "thinking with long videos" via native tool calling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 33816–33826, 2026.

[49] Zuhao Yang, Yingchen Yu, Yunqing Zhao, Shijian Lu, and Song Bai. Timeexpert: An expert-guided video llm for video temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24286–24296, 2025.

[50] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[51] 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 *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, 2025.

[52] 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*, 2025.

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

[54] 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*, pages 13691–13701. IEEE, 2025.

[55] Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-omni: Towards audio-visual reasoning with temporal alignment across modalities. *arXiv preprint arXiv:2505.17862*, 2025.

In this appendix, we provide additional details on the benchmarks used in our evaluation in Section A, the baseline VideoLLMs in Section B, supplementary experimental results in Section C, formal algorithmic details of V-CAST in Section D, and further qualitative comparisons in Section E.

A Benchmark Details

This section provides a brief overview of the video understanding benchmarks used in our evaluation:

- **MVBench** [22] assesses 20 distinct temporal tasks. It defines temporal tasks through a static-to-dynamic formulation and converts public video annotations into multiple-choice questions. The benchmark covers diverse perspectives from first-person to third-person. It requires models to perform complex reasoning beyond single-frame perception by linking actions and event order across successive frames.
- **LongVideoBench** [43] targets long-context comprehension with 3,763 videos. It features a novel referring reasoning task for precise temporal localization. The dataset includes original or transcribed subtitles. This allows for the evaluation of interleaved multimodal inputs across 17 categories, linking visual events to relevant subtitles over long contexts.
- **MLVU** [54] includes videos lasting 3 minutes to 2 hours. It evaluates 9 tasks such as action counting and temporal ordering. The benchmark requires models to process global summaries and fine-grained details. It highlights the challenges of maintaining long-term dependencies when relevant evidence is distributed across different parts of a long video.
- **VideoMME** [12] consists of 900 high-quality videos across 6 primary domains. It integrates audio tracks and subtitles to test comprehensive multimodal perception. The questions cover 30 fine-grained subfields. Content is categorized into short, medium, and long subsets for multi-scale analysis and comparisons across video durations within diverse domains.
- **EgoSchema** [30] is a diagnostic benchmark for long-form egocentric video understanding. It contains over 5,000 human-curated multiple-choice questions derived from the Ego4D dataset. Each task involves a three-minute video clip with long temporal context. It evaluates reasoning over complex human activities from a first-person perspective in daily life.

We further introduce audio-visual video understanding benchmarks in our further evaluation:

- **WorldSense** [15] evaluates omni-modal understanding with 1,662 videos across 8 domains. It strongly couples visual, audio, and text inputs to test synergistic perception. The benchmark contains 3,172 questions covering 26 distinct tasks. It requires models to construct coherent contexts from complex multimodal streams for video question answering.
- **AV-SpeakerBench** [31] evaluates speaker-centric audiovisual reasoning in multi-speaker scenarios. It contains 3,212 questions across 11 perception tasks. The benchmark employs a cross-modal anchor-target structure to test audio-visual alignment. It requires models to jointly track speaker identity, content, and timing to prevent unimodal shortcuts.
- **AVUTBenchmark** [47] targets audio-centric video comprehension with 2,662 videos. It features over 13,000 QA pairs focusing on auditory content and audio-visual interactions. The benchmark employs an answer permutation mechanism to address the "text shortcut" problem. It requires models to genuinely process multimodal signals to ground answers.
- **OmniVideoBench** [21] evaluates synergistic audio-visual intelligence emphasizing modality complementarity. It comprises 1,000 QA pairs from 628 videos spanning up to 30 minutes. The questions cover 13 reasoning types annotated with step-by-step multimodal traces. It requires models to synthesize complex interactions for consistency across modalities.
- **Daily-Omni** [55] targets cross-modal temporal alignment in daily scenarios. It contains 1,197 questions derived from 684 videos across 6 task families. The benchmark explicitly tests the synchronous processing of audio and visual streams. It requires models to perform deep temporal integration across modalities to link visual events with concurrent sounds.

B Model Details

This section summarizes the VideoLLMs evaluated in our experiments:

- **LLaVA-OneVision** [20] establishes a unified framework for single-image, multi-image, and video tasks. It treats videos as interleaved sequences of visual tokens. This design facilitates seamless knowledge transfer from image-based pre-training to video understanding. The model demonstrates exceptional zero-shot performance across diverse temporal tasks.
- **LLaVA-Video** [53] initializes from the LLaVA-OneVision image-stage checkpoints. It undergoes fine-tuning on the LLaVA-Video-178K dataset. This large-scale synthetic corpus covers detailed captioning and complex reasoning. By integrating the SigLIP vision encoder with the Qwen2 language model, it achieves outstanding performance across benchmarks.
- **Qwen3-VL** [2] represents the latest advancement in the Qwen-VL series. It enhances the visual perception system to support higher-resolution inputs and more efficient token compression. The model features an upgraded LLM backbone with improved reasoning logic and multilingual capabilities. It excels at fine-grained visual grounding and complex instruction following in long-duration video streams. MRoPE encodes time and space.
- **Qwen2.5-Omni** [45] introduces a natively multimodal architecture built upon the Qwen2.5 large language model. It supports the seamless integration of text, audio, images, and videos as interleaved inputs. By employing a unified token space and end-to-end joint training, the model effectively captures complex cross-modal alignments. It demonstrates exceptional capabilities in omni-modal reasoning and real-time conversational tasks with audio output.

C More Experimental Results

Performance under Higher Retention Ratio. Table 9 evaluates V-CAST under a larger budget of $R = 35\%$. V-CAST achieves an average score of 65.8 (98.2%) with 64 frames and 63.5 (98.8%) with 32 frames, showing that curvature-aware selection still identifies critical tokens as the budget increases. While FlashVID encounters OOM at 64 frames, V-CAST enables stable inference across sequence lengths due to its attention-free pruning. V-CAST consistently matches or surpasses the strongest baseline, suggesting that routing tokens toward semantic transitions is more effective than global frame uniqueness for preserving spatio-temporal continuity under constrained budgets.

Table 9: **Performance comparison with other baselines on Qwen3-VL-8B-Instruct under a retention ratio of** 35%**.** "Average" denotes the mean performance across benchmarks.

Methods	MVBench	LongVideoBench	MLVU	VideoMME				Average	
				Overall	Short	Medium	Long	Score	%
Max Input Frames=64									
Qwen3-VL-8B-Instruct	69.2	62.8	68.9	66.9	78.8	66.2	55.8	67.0	100.0
Retention Ratio=35%									
VisionZip (CVPR'25)	65.7	58.9	64.3	64.3	74.9	62.2	**55.9**	63.3	94.5
VidCom^2 (EMNLP'25)	**68.9**	61.0	65.5	**66.0**	77.3	64.9	55.8	65.4	97.6
FastVID (NeurIPS'25)	66.5	60.7	66.1	63.0	74.2	60.3	54.4	64.1	95.7
HoliTom (NeurIPS'25)	65.8	60.4	64.4	64.0	75.8	61.8	54.3	63.6	95.0
FlashVID (ICLR'26)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	-	-
V-CAST (Ours)	68.5	**61.6**	**67.1**	**66.0**	**77.6**	**65.6**	55.0	**65.8**	**98.2**
Max Input Frames=32									
Qwen3-VL-8B-Instruct	68.6	60.3	63.5	64.5	76.0	60.4	57.0	64.2	100.0
Retention Ratio=35%									
VisionZip (CVPR'25)	64.4	56.8	61.3	60.8	70.9	57.0	54.6	60.8	94.8
VidCom^2 (EMNLP'25)	68.1	58.6	62.0	62.9	73.6	59.9	55.1	62.9	98.0
FastVID (NeurIPS'25)	67.5	58.7	62.3	61.9	73.1	59.1	53.6	62.6	97.5
HoliTom (NeurIPS'25)	65.0	57.8	62.0	61.3	71.9	57.9	54.0	61.5	95.8
FlashVID (ICLR'26)	67.9	**59.2**	62.4	62.7	**75.2**	59.6	53.3	63.1	98.3
V-CAST (Ours)	**68.3**	58.5	**63.1**	**63.9**	74.7	**60.8**	**56.3**	**63.5**	**98.8**

Performance on Audio-Visual Video Understanding. Table 10 further evaluates V-CAST on Qwen2.5-Omni-7B under the 768-frame audio-visual setting. Although all baselines only compress video tokens, V-CAST achieves the best average score of **49.6** and preserves **97.1%** of the original omni-model performance, outperforming the second-best VidCom^2 [26] by **+0.1** points. V-CAST also achieves the lowest total inference time of **19,170.8 s**, a **50.0%** reduction relative to the uncompressed model. These results indicate that the proposed temporal routing and spatial token selection remain effective when visual evidence must be integrated with audio cues for omni-video reasoning.

Table 10: **Performance comparison with other baselines on Qwen2.5-Omni-7B across omni-benchmarks.** We use the 768-frame input setting.

Methods	WorldSense	AV-Speaker Bench	AVUT	OmniVideo Bench	Daily-Omni	Average ↑ Score	Average ↑ %	Total Inference Time (s) ↓
Qwen2.5-Omni-7B	45.6	47.2	64.6	34.2	63.7	51.1	100.0	38,357.6
Retention Ratio=25%								
VisionZip (CVPR'25)	45.0	42.5	62.2	34.4	59.5	48.7	95.4	23,384.0 (↓39.0%)
VidCom^2 (EMNLP'25)	45.3	43.2	62.3	**35.6**	**61.1**	49.5	96.9	19,746.1 (↓48.5%)
FastVID (NeurIPS'25)	**45.6**	**44.1**	62.7	34.4	59.7	49.3	96.6	20,557.8 (↓46.4%)
HoliTom (NeurIPS'25)	**45.6**	43.0	61.9	34.7	58.3	48.7	95.4	26,898.8 (↓29.9%)
V-CAST (Ours)	45.2	43.9	**63.5**	**35.6**	59.6	**49.6**	**97.1**	**19,170.8** (↓50.0%)

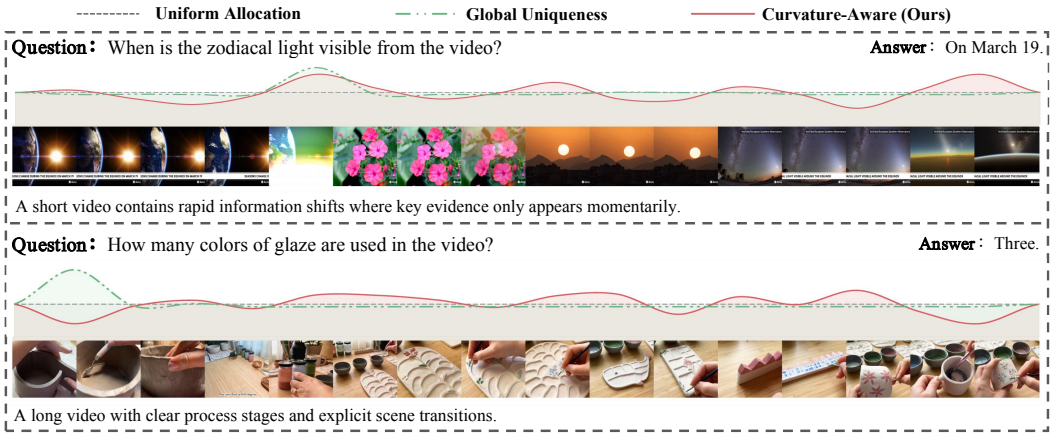


Figure 7: **Visualization of frame budget allocation.** We compare Uniform allocation, Global Uniqueness (VidCom^2 [26]), and Curvature-Aware allocation (V-CAST) at $R = 25\%$. A higher curve indicates a larger per-frame token budget.

Case Analysis of Temporal Budget Allocation. Figure 7 provides qualitative evidence for how different temporal allocation rules distribute token budgets over time. In the first example, the question asks when the zodiacal light is visible, and the key evidence appears only briefly around the transition from generic sky or seasonal frames to the zodiacal-light observation frames. Uniform allocation keeps a nearly flat budget and cannot emphasize this short evidence window, while Global Uniqueness assigns a strong peak to a globally distinctive segment but does not consistently follow all local transitions. By contrast, V-CAST produces multiple targeted peaks around the changing visual states, including the final observation interval that supports the answer. In the second example, the video contains a staged glazing process, where the answer depends on accumulating evidence across several process changes rather than a single distinctive frame. Global Uniqueness over-emphasizes the early visually distinctive frames, whereas V-CAST allocates budget to later transitions such as painting, color application, and object close-ups. These cases illustrate that curvature-aware allocation is not simply selecting globally unusual frames; it routes temporal density toward local representation changes, which better matches short evidence windows and multi-stage video processes.

Broad Applicability of Curvature-Aware Budgeting. Figure 8 shows that our curvature-aware budgeting can be readily and effectively integrated into other compression methods. Under the same $R = 25\%$, adding curvature-aware budgeting consistently improves the performance of VisionZip [46] and VidCom^2 [26] across different tasks. The gains are more evident for VisionZip, suggesting that adaptive temporal budget allocation is especially beneficial when the original method lacks explicit temporal importance modeling. For VidCom^2, the improvements are smaller but consistent, as expected since it already accounts for frame unique-

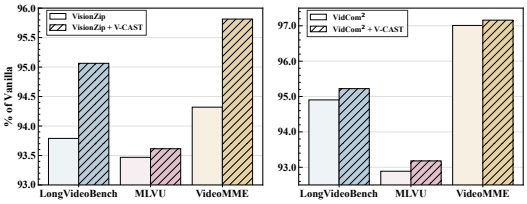


Figure 8: **Broad applicability of V-CAST budgeting on other frame compression methods.** "+V-CAST" denotes equipping the original method with our curvature-aware budgeting strategy.

ness during compression. These results suggest that our budgeting is broadly compatible with different compression paradigms and can serve as a generic temporal allocation strategy.

Hyper-parameter Sensitivity Analysis. We further analyze the sensitivity of V-CAST to two key hyper-parameters in the main experiments: the temporal budget temperature τ (default: 0.7) and the spatial anchor weights (default: 1.0, 1.0). Table 11 (a) shows that V-CAST remains stable across τ choices, with the average score varying only from 65.0 to 65.3 as τ changes from 0.6 to 0.9. Among the tested values, $\tau = 0.7$ gives the best trade-off, achieving the highest VideoMME overall, short, medium, and long scores, while matching the best MLVU result up to rounding. In contrast, a larger temperature ($\tau = 0.9$) makes the allocation more uniform and slightly weakens focus on decisive frames. Overall, a moderate temperature best balances emphasis on critical moments and temporal context. Table 11 (b) shows that V-CAST remains robust under different anchor weight settings. The balanced choice $(1.0, 1.0)$ achieves the best average score of 65.3, suggesting that the two anchors provide complementary cues for spatial selection. Overall, V-CAST is not highly sensitive to the exact weight choice, while a balanced configuration yields the most consistent performance.

Table 11: **Sensitivity analysis on V-CAST hyper-parameters.** We evaluate (a) temporal temperature τ and (b) spatial selection weights on Qwen3-VL-8B-Instruct at $R = 25\%$.

(a) Ablation on Temperature τ.

τ	MLVU	VideoMME				Avg Score
		Overall	Short	Medium	Long	
Vanilla	68.9	66.9	78.8	66.2	55.8	67.9
0.6	65.3	64.8	76.2	63.1	55.1	65.1
0.7	**65.4**	**65.1**	**76.8**	**63.2**	**55.3**	**65.3**
0.9	**65.4**	64.6	76.1	62.8	55.0	65.0

(b) Ablation on Anchor Weights.

Weight	MLVU	VideoMME				Avg Score
		Overall	Short	Medium	Long	
Vanilla	68.9	66.9	78.8	66.2	55.8	67.9
(1.3, 0.7)	65.0	64.9	76.0	63.0	**55.7**	65.0
(1.1, 0.9)	65.0	64.7	75.6	**63.2**	55.2	64.9
(1.0, 1.0)	65.4	**65.1**	76.8	**63.2**	55.3	**65.3**
(0.9, 1.1)	**65.7**	64.7	**77.0**	62.6	54.7	65.2

D Algorithm Details of V-CAST

Algorithm 1 summarizes V-CAST. It first allocates the global token budget across frames according to semantic curvature, and then selects the top-b_t spatial tokens in each frame using the dual-anchor diversity-saliency score. The selected tokens retain their original spatio-temporal coordinates.

E More Qualitative Comparisons

Figure 9 presents additional qualitative comparisons among V-CAST, various token compression baselines, and the vanilla VideoLLM. These cases cover a diverse range of challenging video understanding scenarios, including rapid action shifts, subtle semantic transitions, and complex long-term reasoning. The examples show that V-CAST more effectively identifies and preserves brief but decisive moments by dynamically routing token budgets to critical event boundaries. Unlike existing methods that often dilute fleeting evidence under tight budgets, V-CAST highlights and preserves these task-critical moments. Consequently, V-CAST yields correct answers in complex cases where conventional compression baselines, and sometimes even the vanilla model, fail to capture the decisive visual cues. These cases highlight the role of temporal coverage in reasoning.

Algorithm 1 V-CAST: Curvature-Guided Spatio-Temporal Token Pruning

Require: Visual tokens $\mathbf{X} = \{\mathbf{x}_{t,i}\} \in \mathbb{R}^{T \times P \times D'}$, retention ratio r, temperature τ, minimum per-frame budget $k_{\min}$

Ensure: Pruned visual token sequence $\tilde{\mathbf{X}}^{\flat}$

$B \leftarrow \lfloor rTP \rfloor$

for $t = 1, \ldots, T$ **do**

  $\bar{\mathbf{x}}_t \leftarrow \frac{1}{P} \sum_{i=1}^{P} \mathbf{x}_{t,i}, \quad \hat{\mathbf{x}}_t \leftarrow \bar{\mathbf{x}}_t / (\|\bar{\mathbf{x}}_t\|_2 + \epsilon)$

end for

for $t = 1, \ldots, T$ **do**

  $\Delta_t \leftarrow \hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t$

  $\kappa_t \leftarrow 1 - \cos(\Delta_{t-1}, \Delta_t)$ ▷ boundary indices are clamped

end for

$p_t \leftarrow \exp(\kappa_t / \tau) \big/ \sum_{j=1}^{T} \exp(\kappa_j / \tau)$

Convert $\{p_t\}_{t=1}^{T}$ into integer budgets $\{b_t\}_{t=1}^{T}$ such that $\sum_t b_t = B$ and $b_t \in [k_{\min}, P]$

for $t = 1, \ldots, T$ **do**

  for $i = 1, \ldots, P$ **do**

    $s_{t,i} \leftarrow \big(1 - \cos(\mathbf{x}_{t,i}, \bar{\mathbf{x}}_t)\big) + \mathcal{N}(\|\mathbf{x}_{t,i}\|_2)$

  end for

  $\mathcal{S}_t \leftarrow \mathrm{TopB}\left(\{\mathbf{x}_{t,i}\}_{i=1}^{P}, \{s_{t,i}\}_{i=1}^{P}, b_t\right)$

end for

return $\tilde{\mathbf{X}}^{\flat} \leftarrow \mathrm{Concat}(\mathcal{S}_1, \ldots, \mathcal{S}_T)$



User **Q:** What was Amazon's stock price at the peak of the dot-com bubble?
"A. $7/Share." "B. $100/Share.",
"C. $600/Share." "D. $300/Share."

Qwen3-VL Vanilla "C.." | VisionZip "C.." | FastVID "C.." | FlashVID "C.." | VidCom² "C.." | **V-CAST** "B.."



User **Q:** What is the pregnant woman wearing in the video?
"A strapless, form-fitting dress adorned with sequins."
"B. A mint green dress with a strapless design."
"C. An elegant black dress with long sleeves."
"D. A striking green dress with a large, ruffled detail."

Qwen3-VL Vanilla "A.." | VisionZip "B.." | FastVID "B.." | FlashVID "A.." | VidCom² "C.." | **V-CAST** "C.."



User **Q:** Which training exercise in the video does the man perform with the assistance of an elastic rope?
"A. Pull-ups with the assistance of an elastic rope.",
"B. Lying triceps extension with the assistance of an elastic rope.",
"C. Sit-ups with the assistance of an elastic rope.",
"D. Push-ups with the assistance of an elastic rope."

Qwen3-VL Vanilla "B.." | VisionZip "B.." | FastVID "B.." | FlashVID "B.." | VidCom² "B.." | **V-CAST** "A.."



User **Q:** How many people are on each team in the video?
"A. 6." "B. 4." "C. 5." "D. 3."

Qwen3-VL Vanilla "B.." | VisionZip "B.." | FastVID "B.." | FlashVID "B.." | VidCom² "B.." | **V-CAST** "D.."



User **Q:** What treat did the little hamster savor at the end of the video?
"A. Salmon." "B. Bacon." "C. Scallop." "D. Kobe beef."

Qwen3-VL Vanilla "B.." | VisionZip "B.." | FastVID "B.." | HoliTom "B.." | VidCom² "B.." | **V-CAST** "D.."



User **Q:** Which countries do the top three in the competition come from?
"A. Qatar, South Korea, Ukraine."
"B. United States, South Korea, Italy."
"C. Ukraine, Qatar, United States." "D. Italy, Qatar, Ukraine."

Qwen3-VL Vanilla "D.." | VisionZip "D.." | FastVID "B.." | FlashVID "D.." | VidCom² "D.." | **V-CAST** "A.."

Figure 9: **Additional qualitative comparisons.** V-CAST preserves task-critical moments and correctly answers challenging cases where baseline compression methods, and occasionally even the vanilla model, miss decisive visual evidence.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction match the core claims of V-CAST, including training-free video token pruning, spatio-temporal evidence coverage, and the reported accuracy–efficiency trade-offs. The claims do not assert results beyond the evaluated VideoLLM backbones, benchmarks, and ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: The current draft does not include a dedicated limitations discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper provides theoretical intuition and derivations for the proposed pruning rule, but does not present formal theorems or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies the evaluated VideoLLM backbones, benchmarks, frame settings, retention ratios, baselines, evaluation framework, hardware, and the main ablation factors used to obtain the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material provides the V-CAST code and reproduction instructions, and the experiments use publicly available benchmarks, models, and evaluation tools. The paper also describes the algorithm and experimental settings needed to reproduce the main results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: V-CAST is training-free, so optimizer and training details are not required. The experimental section specifies the benchmarks, evaluation framework, VideoLLM backbones, input-frame settings, retention ratios, compared methods, hardware, and ablation factors.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports benchmark results and ablations across multiple models, datasets, frame settings, and retention ratios, but does not yet include repeated runs, confidence intervals, or formal significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper states that experiments use NVIDIA H100-80GB HBM3 GPUs and reports latency, throughput, and peak memory for the main efficiency setting, but does not yet quantify GPU hours, wall-clock cost, or total project compute for every experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research concerns inference-time efficiency for existing VideoLLMs using public benchmarks and does not involve human subjects, private data collection, or unsafe intervention protocols.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: This work focuses on improving inference-time efficiency for VideoLLMs through visual token pruning. While such techniques may improve deployment efficiency and accessibility, the current paper does not include a dedicated discussion of broader positive or negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not release a new foundation model, image generator, or scraped dataset; it releases an inference-time pruning method for existing VideoLLMs.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper cites the external benchmarks, VideoLLM backbones, baseline methods, and evaluation framework used in the experiments, but the current draft does not yet provide a consolidated license table for all external models, datasets, and code assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new released dataset or model asset. The V-CAST code is provided in the supplementary material as reproduction material for the proposed method.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not involve crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve human subjects and therefore does not require IRB review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a non-standard component of the proposed method; the paper studies inference-time visual token pruning for existing VideoLLM backbones.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.