MoPrune: Scene-Guided Motion-Aware Token Pruning for Efficient Video Large Language Models

Anonymous ACL submission

Abstract

Video Large Language Models (VideoLLMs) struggle with the heavy computational cost of long or high-resolution videos due to massive visual token counts and the quadratic complexity of attention. Prior pruning approaches mainly rely on token importance or similarity, while largely overlooking video dynamics and the fact that different scenes exhibit different redundancy patterns. We introduce MoPrune, a training-free, scene-guided and motion-centric token pruning framework for accelerating VideoLLMs. MoPrune first segments videos into semantically coherent scenes to preserve temporal and motion consistency. Within each scene, it determines frame retention rates from intra-scene frame uniqueness. Finally, at the token level, MoPrune retains visually distinctive tokens and motion-salient tokens via a unified score, preserving both informative static details and dynamic regions. Extensive experiments across multiple VideoLLMs and public benchmarks demonstrate MoPrune's superior efficiency–performance trade-offs. On LLaVA-OneVision, retaining 25% of visual tokens matches or slightly improves the dense baseline, and retaining 15% tokens preserves 99% of the original performance. MoPrune is fully compatible with hardware-efficient techniques such as Flash Attention.

1 Introduction

Video large language models (VideoLLMs) have recently achieved significant performance on many video understanding tasks, including spatiotemporal perception, reasoning and instruction following (Li et al., 2025a; Zhang et al., 2024b; Li et al., 2024a,b). However, unlike single images, videos consist of many consecutive frames, which leads to an explosion in the number of visual tokens—often dominating the input length of the underlying LLM (Xu et al., 2024a; Chen et al., 2024b). Furthermore, the increasing application demands requiring

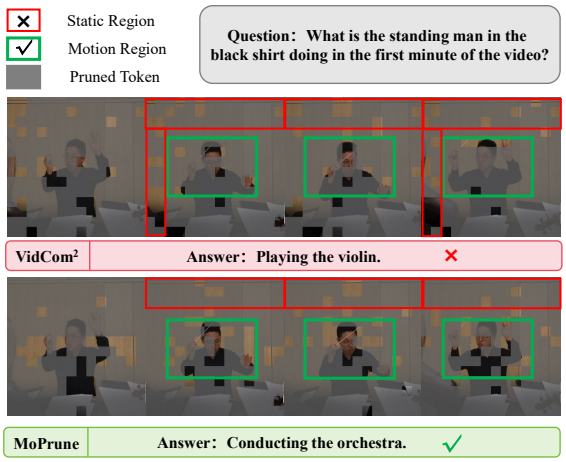


Figure 1: **Effect of motion aware.** Without sufficient motion perception, video understanding can sometimes be inaccurate, since motion tokens offer richer and less redundant information for VideoLLMs.

VideoLLMs to process longer and more complex videos can lead to excessive computational overhead and memory consumption (Xu et al., 2024b; Liu et al., 2025c).

To reduce this overhead, token pruning has been proposed to accelerate VideoLLMs inference by removing redundant visual information (Fu et al., 2025b; Huang et al., 2025; Tao et al., 2025). Existing approaches can be divided into pre-LLM (Zhang et al., 2025c) and intra-LLM (Chen et al., 2024a). In the pre-LLM stage, some existing methods rely on computationally intensive clustering algorithms or complex information maximization planning (Alvar et al., 2025; Sun et al., 2025; Huang et al., 2025), which introduces considerable computational overhead. Furthermore, most strategies apply a uniform compression strategy to all frames, but the information richness of each frame is different. VidCom^2 (Liu et al., 2025b) adaptively adjusts the compression intensity across frames by quantifying frame uniqueness, but it does not consider scene transitions; consequently, frames from a visually distinctive scene can dominate the global

budget in a way that is suboptimal for multi-scene videos.

A second limitation is that most pruning criteria emphasize appearance similarity or token "importance" while under-utilizing *motion*, which is central to video understanding. Intuitively, tokens undergoing large temporal changes are more likely to carry the evidence needed for temporal reasoning (Zhao et al., 2025; Liu et al., 2025a), yet informative static details should also be preserved.

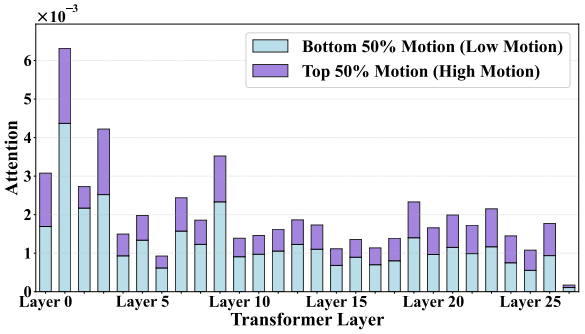
Empirically, we observe a clear motion–attention correlation in VideoLLMs. Using LLaVA-Video-7B (Zhang et al., 2024b), we estimate motion magnitude via per-token feature differences between consecutive frames and split tokens into high- and low-motion halves. Across MVBench (Li et al., 2024c) and VideoMME (Fu et al., 2025a), high-motion tokens consistently receive larger decoder attention throughout generation (summing attention weights over all heads per token and averaging within each group). This suggests that motion-agnostic pruning can disproportionately remove critical evidence, motivating motion-aware pruning that preserves dynamic regions while retaining informative static context.

Intra-LLM pruning methods require explicit attention weights at specific LLM layers (Fu et al., 2025b; Zhang et al., 2024a; Xing et al., 2024), making them incompatible with efficient attention kernels (e.g., FlashAttention (Dao et al., 2022)) and may increase peak memory.

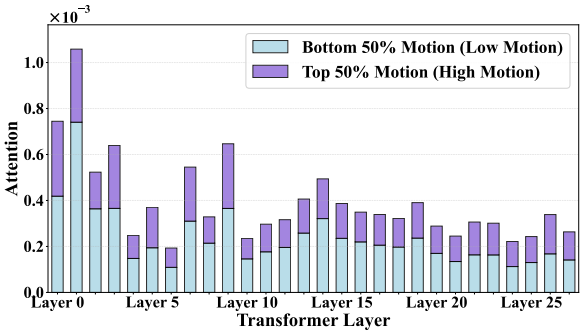
To address this, we propose MoPrune, which measures the pruning strength across different frames within a scene and performs token selection based on token saliency and motion amplitude. We apply MoPrune to LLaVA-Video (Zhang et al., 2024b) and LLaVA-OneVision (Li et al., 2024a), and evaluate it on multiple video question-answering benchmarks. Results show that MoPrune substantially reduces inference cost while preserving (and sometimes slightly improving) accuracy.

The main contributions are below:

- We reformulate video token pruning from a video dynamics perspective, modeling it as a problem of maximizing scene-level saliency and motion capture.
- We propose MoPrune, a training-free, scene-guided pruning framework that adaptively allocates token budgets across scenes and frames, and selects tokens based on both scene-/frame-level uniqueness and motion magnitude.
- Extensive experiments demonstrate that MoPrune achieves superior accuracy–efficiency trade-offs; e.g., on LLaVA-OneVision, retaining 25% tokens matches or slightly improves the dense baseline, while 15% tokens preserves 99% performance.



(a) MVBench



(b) VideoMME

Figure 2: Motion-salient tokens receive higher decoder attention. On LLaVA-Video-7B, we split tokens by motion magnitude (top/bottom 50%) and average per-token attention (summed over heads) per layer; the trend holds on (a) MVBench and (b) VideoMME, motivating motion-aware pruning.

2 Related Work

2.1 Video Large Language Models

VideoLLMs use a visual encoder to encode each frame of a video individually into visual tokens, which are then fed into the LLM independently through a projection layer along with the user query embeddings. Recent research has advanced video understanding to a deeper level (Song et al., 2024; Zhang et al., 2025a; Chen et al., 2024c). Qwen2-VL (Wang et al., 2024) uses M-RoPE to enhance temporal awareness. LLaVA-OneVision (Li et al., 2024a) unifies image and video tasks and efficiently compresses tokens through bilinear interpolation. LLaVA-Video (Zhang et al., 2024b) uses newline

tokens for spatiotemporal grounding. However, the long sequences of visual tokens from consecutive video frames limit their practical applications, and the long duration and high resolution requirements of complex video understanding tasks lead to significant computational overhead.

2.2 Token Compression for LVLMs

Token compression improves the inference efficiency of VLLMs by reducing visual tokens. An increasing number of methods are shifting from perceptual training (Li et al., 2025b) to training-free methods (Chen et al., 2024a). FastV (Chen et al., 2024a) first identifies inefficient cross-modal attention in language models and evaluates the importance of visual tokens based on attention received from text tokens. VisionZip (Yang et al., 2025) uses attention to select domain tokens using [CLS] tokens in the visual encoder and then merges contextual tokens. However, these methods are not designed for video understanding, treating video frames as independent images and ignoring temporal relationships within the video. Recent VideoLLM-specific methods like DyCoke (Tao et al., 2025) selectively prune by merging redundant tokens every four consecutive frames and applying a dynamic key-value buffer. FrameFusion (Fu et al., 2025b) finds that token similarity is highly consistent across different layers of the LLM, so it computes the similarity of corresponding positions in adjacent frames in the initial consecutive layers to apply token merging, and then utilizes attention pruning in deeper layers. VidCom^2 (Liu et al., 2025b) quantizes the uniqueness of frames using a global token representation of the video and combines it with intra-frame token uniqueness for token selection. However, all these methods ignore the uniqueness of tokens within the scene and the magnitude of motion. To address this, we propose a training-free video trimming strategy that performs scene-independent large motion amplitude or strong uniqueness token preservation, thereby more effectively reducing computational resources while maintaining overall performance.

3 Methodology

3.1 Preliminary

A typical VideoLLM consists of three components: a visual encoder, a modal projection layer, and the LLM (Li et al., 2024a; Zhang et al., 2024b). Given a video sequence $\mathbf{V} = \{\mathbf{v}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times H\times W\times 3}$, it is first converted into visual tokens by a pre-trained image encoder. The projection layer aligns these visual tokens with the word embedding space of the LLM, and then feeds them into the LLM for autoregressive response generation.

Token pruning aims to directly compress the number of tokens to accelerate inference. Essentially, it selects the optimal subset from a given set of visual tokens under a specific retention rate constraint. Considering that videos often contain multiple scenes with different semantics and redundancy patterns, the representativeness of tokens should be preserved under scene-independent conditions. Our goal is to design an adaptive tuning and evaluation mechanism that minimizes redundancy while maximizing model performance preservation.

3.2 MoPrune

We propose MoPrune, a training-free, pre-LLM pruning framework that is both scene-aware and motion-aware. MoPrune reduces redundancy while remaining compatible with hardware-efficient attention implementations (e.g., FlashAttention).

As illustrated in Figure 3, MoPrune operates in two stages. First, it partitions a video into semantically coherent scenes and allocates frame-level token budgets based on scene-relative frame uniqueness. Second, it selects the final set of tokens for each frame using a unified score that combines appearance distinctiveness with motion magnitude. We describe these components in detail below.

3.3 Scene Segmentation and Adaptive Frame Adjustment

Scene Segmentation. A video often contains multiple scenes with different semantics and redundancy patterns. Treating all frames as a single token pool can mix unrelated content and can also corrupt motion estimation at abrupt scene changes. We therefore first partition the input video into K semantically coherent scenes along the temporal axis.

For each frame t, we compute a global representation by average-pooling its visual tokens $\mathbf{x}_t \in \mathbb{R}^{N\times D}$:

$$\mathbf{g}_t = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_{t,n}, \quad \mathbf{g}_t \in \mathbb{R}^D. \tag{1}$$

We then compute cosine similarity between adjacent frames,

$$s_t = \frac{\mathbf{g}_t \cdot \mathbf{g}_{t+1}}{\|\mathbf{g}_t\|\,\|\mathbf{g}_{t+1}\|}, \quad s_t \in [-1, 1], \tag{2}$$

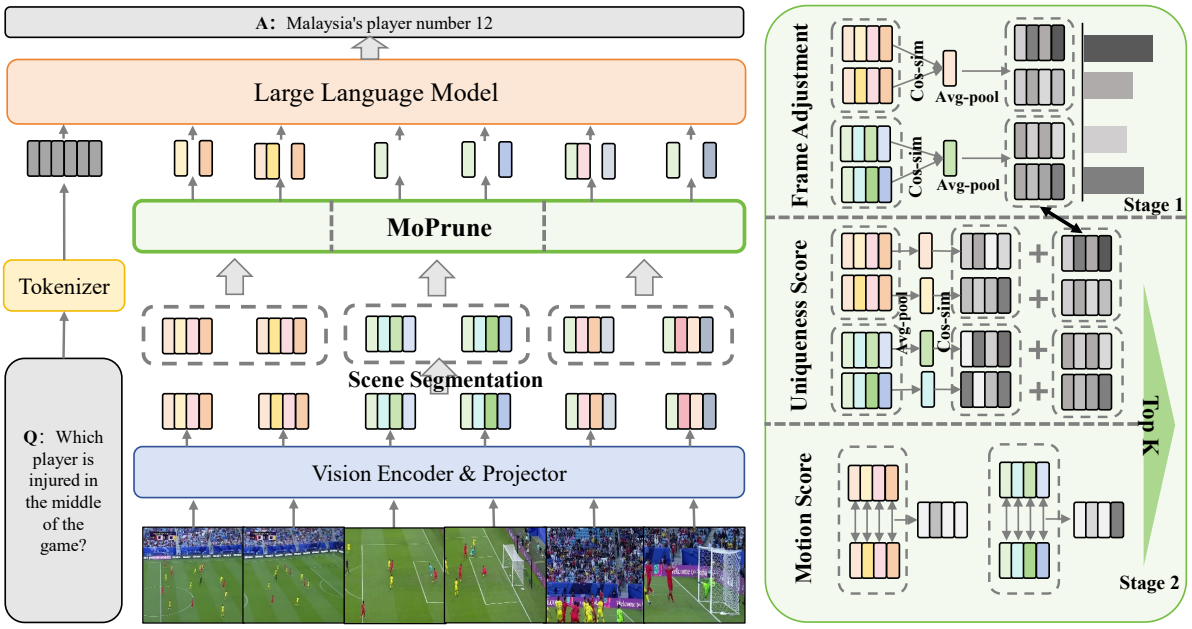


Figure 3: **Overall framework of MoPrune.** Our MoPrune performs plug-and-play token compression in two stages: (i) Scene Segmentation and Adaptive Frame Adjustment, (ii) Uniqueness and Motion-Aware Token Pruning.

and detect a scene boundary at frame t if $s_t < \tau$. This yields a set of scenes $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K\}$, where each $\mathcal{S}_k$ is a temporally contiguous set of frames.

Given a global token retention ratio $r \in (0, 1]$, we allocate a token budget to each scene proportional to its length: $B_k = r \cdot T_k \cdot N$, where $T_k = |\mathcal{S}_k|$.

Adaptive Frame Adjustment. Within each independent scene, different frames have varying levels of information richness and should be given different levels of attention. VidCom^2 (Liu et al., 2025b) demonstrates that simulating human visual perception should naturally focus on unique frames, thus designing a method to quantify frame uniqueness.

VidCom^2 (Liu et al., 2025b) estimates frame uniqueness at the video level, but multi-scene videos with large distribution shifts can make this metric overly coarse and less sensitive to fine-grained differences among frames within the same scene; moreover, global "rarity" may be dominated by scene changes or visually atypical yet uninformative frames. We therefore compute scene-relative frame uniqueness and use it to adaptively adjust per-frame retention within each scene, which suppresses redundant frames and better highlights truly distinctive details, yielding a more structured token allocation (Figure 4).

Specifically, for each scene $\mathcal{S}_k$, we compute a scene centroid feature by averaging all tokens in that scene:

$$\mathbf{g}_k = \frac{1}{T_k \cdot N} \sum_{t \in \mathcal{S}_k} \sum_{n=1}^{N} \mathbf{x}_{t,n}, \quad \mathbf{g}_k \in \mathbb{R}^D. \tag{3}$$

Where T_k represents the number of frames in scene S_k. Then we calculate the similarity between each token within the scene and the corresponding global representation of the scene:

$$s_{t,n}^{\text{scene}} = \frac{\mathbf{x}_{t,n} \cdot \mathbf{g}_k}{\|\mathbf{x}_{t,n}\| \, \|\mathbf{g}_k\|}, t \in \mathcal{S}_k, s_{t,n}^{\text{scene}} \in [-1, 1]. \tag{4}$$

Lower $s_{t,n}^{\text{scene}}$ indicate greater uniqueness within the scene. The frame uniqueness score is defined as $u_t = \frac{1}{N} \sum_{n=1}^{N} -s_{t,n}^{\text{scene}}$. A larger u_t indicates that the frame t occupies a more special position within the scene. We compute $\tilde{u}_t = (u_t - \max(u_t))/\alpha$ ($\alpha = 0.01$), and obtain the relative importance weight within the scene via softmax:

$$\sigma_t = \frac{\exp(\tilde{u}_t)}{\sum_{j \in \mathcal{S}_k} \exp(\tilde{u}_j) + \epsilon}. \tag{5}$$

The base retention rate of a frame in each scene is: $\gamma_k^{\text{base}} = \frac{B_k}{T_k}$. Then, the base retention rate of the frame is adjusted using importance weight according to the following formula:

$$\gamma_t = \gamma_k^{\text{base}} \cdot \left(1 + \sigma_t - \frac{1}{T_k}\right), t \in \mathcal{S}_k. \tag{6}$$

This achieves dynamic adjustment of frames under scene-independent conditions, realizing differentiated retention levels.

Figure 4: **Per-frame token retention ratio for different video sequences.** Only a contiguous subset of frames from each video is visualized. ■ indicates frame retention without scene segmentation, while ■ indicates frame retention with scene segmentation.

3.4 Uniqueness and Motion-Aware Token Pruning

Uniqueness Score. To cover as much of the entire video as possible, we use a uniqueness metric, similar to VidCom^2 (Liu et al., 2025b), to retain more unique information. However, we evaluate scene-level and frame-level uniqueness instead of the video-level. First, for frame t, we use the previously obtained frame-level global representation $\mathbf{g}_t$ to calculate the similarity between each token within the frame and the global representation, as follows:

$$s^{\text{frame}}_{t,n} = \frac{\mathbf{x}_{t,n} \cdot \mathbf{g}_t}{\|\mathbf{x}_{t,n}\| \, \|\mathbf{g}_t\|}, s^{\text{frame}}_{t,n} \in [-1, 1]. \quad (7)$$

Then we obtain the frame-level uniqueness score $u^{\text{frame}}_{t,n} = -s^{\text{frame}}_{t,n}$. For the scene-level score, we use the previously calculated $u^{\text{scene}}_{t,n} = -s^{\text{scene}}_{t,n}$. We add these two scores together to get the total uniqueness score:

$$u_{t,n} = u^{\text{scene}}_{t,n} + u^{\text{frame}}_{t,n}. \quad (8)$$

This score assigns higher values to tokens that are outliers both within their local frame context and within the broader scene context.

Motion Score. To capture temporal dynamics in video sequences, we compute a motion score for each token by measuring feature-space differences between consecutive frames. This motion score quantifies how much each spatial location changes across time, which is crucial for identifying dynamic regions that contain important temporal information. For each token $\mathbf{x}_{t,n}$ at frame t and spatial position n, we compute its motion score as the L2 norm of the feature difference between consecutive frames:

$$m_{t,n} = \|\mathbf{x}_{t,n} - \mathbf{x}_{t-1,n}\|_2 \quad (9)$$

For the first frame of each scene, we set the motion score to zero, as there is no previous frame within the same scene for comparison. A higher motion score indicates that the corresponding spatial location exhibits significant changes between consecutive frames, suggesting it contains dynamic content that may be important for understanding temporal relationships in the video.

We adopt feature-space temporal difference as our default motion estimator due to its simplicity and negligible overhead. For completeness, we also evaluate optical-flow-based and macroblock-based alternatives. Specifically, (i) for dense optical flow, we compute Farnebäck flow between consecutive frames and use the flow magnitude as the motion score; to align with the patch-token grid and reduce cost, we first downsample each frame to the token grid resolution and compute flow on the downsampled frames. (ii) for sparse optical flow, we track a regular grid of points using pyramidal Lucas–Kanade (PyrLK) and use the per-point displacement magnitude as the motion score, which is then reshaped to the token grid. (iii) for macroblock motion, we partition each frame into non-overlapping patch-sized blocks aligned to the patch-token grid and estimate per-block motion via local template matching between consecutive frames.

Methods	MVBench	LongVideoBench	MLVU	VideoMME				Average	
				Overall	Short	Medium	Long	Score	%
LLaVA-OV-7B	58.3	56.4	63.0	58.5	70.2	56.6	48.8	59.1	100.0
Retention Ratio=25%									
FastV [ECCV'24]	55.5	53.3	59.6	55.3	65.0	53.8	47.0	55.9	94.6
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
DyCoke [CVPR'25]	49.5	48.1	55.8	51.0	61.1	48.6	43.2	51.1	86.5
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
VidCom^2 [EMNLP'25]	57.2	55.2	62.8	58.5	69.1	56.9	49.6	58.4	98.8
MoPrune	**57.7**	**56.5**	**63.5**	**59.7**	**70.4**	**58.3**	**50.4**	**59.4**	**100.5**
Retention Ratio=15%									
FastV [ECCV'24]	51.6	48.3	55.0	48.1	51.4	49.4	43.3	50.8	86.0
SparseVLM [ICML'25]	52.9	49.7	57.4	53.4	61.0	52.1	47.0	53.4	90.4
VisionZip [CVPR'25]	55.7	54.2	60.0	55.5	63.8	54.4	48.3	56.4	95.4
PruneVid [ACL'25]	55.0	**55.6**	61.9	56.8	67.9	54.3	48.1	57.3	97.0
FrameFusion [ICCV'25]	55.1	53.0	58.3	55.5	65.8	54.1	46.7	55.5	93.9
VidCom^2 [EMNLP'25]	54.2	52.8	60.9	56.2	65.8	54.8	48.1	56.0	94.8
MoPrune	**57.7**	55.4	**62.3**	**58.5**	**70.1**	**56.3**	**49.0**	**58.5**	**99.0**
LLaVA-Video-7B	60.3	58.8	67.5	64.4	77.3	62.3	53.7	62.8	100.0
Retention Ratio=25%									
FastV [ECCV'24]	53.8	51.2	57.8	59.3	67.1	60.0	50.8	55.5	88.4
SparseVLM [ICML'25]	55.4	54.2	58.9	60.1	71.1	59.1	50.1	57.2	91.1
DyCoke [CVPR'25]	50.8	53.0	56.9	56.1	65.8	53.6	48.9	54.2	86.3
PruneVid [ACL'25]	55.0	57.9	**64.1**	60.5	72.2	58.6	50.7	59.4	94.6
VidCom^2 [EMNLP'25]	57.0	56.0	59.1	61.8	73.3	61.6	50.4	58.5	93.2
MoPrune	**58.7**	**58.3**	60.9	**62.4**	**75.1**	**60.8**	**51.4**	**60.1**	**95.7**
Retention Ratio=15%									
FastV [ECCV'24]	44.0	44.6	53.8	51.3	56.4	51.1	46.2	48.4	77.1
SparseVLM [ICML'25]	53.1	52.7	56.2	55.7	65.0	53.9	48.3	54.4	86.6
PruneVid [ACL'25]	54.0	**56.2**	**63.3**	59.4	71.2	57.7	49.3	58.2	92.7
VidCom^2 [EMNLP'25]	53.4	52.4	56.8	58.3	68.2	56.9	49.8	55.2	87.9
MoPrune	**58.2**	56.1	60.4	**61.4**	**74.1**	**58.9**	**51.1**	**59.0**	**93.9**

Table 1: **Performance comparisons with baseline methods across different benchmarks using LLaVA-OV-7B and LLaVA-Video-7B .** "Average" shows the mean performance across different benchmarks. DyCoke requires pruning similar tokens from consecutive 4 frames, making it not possible for the retention ratio of $R < 25\%$.

Token Pruning. During token selection, we combine the uniqueness scores and motion scores using normalized sum. We normalize both scores to ensure they are on a comparable scale and aligned in the same direction. For each frame, we normalize the combined uniqueness score $u_{t,n}$ and the motion score $m_{t,n}$ to the range $[0, 1]$ within the frame using min-max normalization:

$$u_{t,n}^{\text{norm}} = \frac{u_{t,n} - u_t^{\min}}{u_t^{\max} - u_t^{\min}}, m_{t,n}^{\text{norm}} = \frac{m_{t,n} - m_t^{\min}}{m_t^{\max} - m_t^{\min}} \tag{10}$$

To ensure that selected tokens are both uniqueness and motion, we fuse using $u_{t,n}$ and $m_{t,n}$ using a weighted geometric mean implemented in the log domain:

$$s_{t,n}^{\text{final}} = \exp\big((1-\eta)\log(u_{t,n}+\varepsilon) + \eta\log(m_{t,n}+\varepsilon)\big) \tag{11}$$

where $\eta = 0.5$ (by default) balances the contributions. For each frame t, we select the top-k_t tokens with the highest final scores, where $k_t = \lfloor \gamma_t \cdot N \rfloor$ is determined by the frame retention rate γ_t and N is the number of tokens per frame. This ensures that tokens with high uniqueness and high motion are prioritized for retention, preserving both spatially distinctive and temporally dynamic content in the pruned token set.

4 Experiments

4.1 Experimental Setting

Benchmarks. We use LMMs-Eval (Zhang et al., 2025b) as the performance evaluation framework and conduct comprehensive comparative experiments on multiple benchmarks, including MVBench (Li et al., 2024c), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2024), and

Methods	LLM Generation↓ Latency (s)	Model Generation↓ Latency (s)	GPU Peak↓ Memory (GB)	Throughput↑ (samples/s)	Performance↑
LLaVA-OV-7B	4775.0	7289.7	17.7	0.64	58.3
Retention Ratio=25%					
FastV [ECCV'24]	2015.1 (↓57.8%)	4687.3 (↓35.7%)	24.7 (↑39.5%)	0.83 (1.30×)	55.5 (↓2.8)
SparseVLM [ICML'25]	3170.6 (↓33.6%)	5839.0 (↓19.9%)	27.1 (↑53.1%)	0.67 (1.05×)	56.4 (↓1.9)
DyCoke [CVPR'25]	1585.3 (↓66.8%)	4322.8 (↓40.7%)	16.1 (↓9.0%)	**0.88 (1.38×)**	49.5 (↓8.8)
VidCom^2 [EMNLP'25]	**1291.0 (↓73.0%)**	**3798.4 (↓47.9%)**	**16.0 (↓9.6%)**	**0.88 (1.38×)**	57.2 (↓1.1)
MoPrune	1301.6 (↓72.7%)	3891.2 (↓46.6%)	**16.0 (↓9.6%)**	**0.88 (1.38×)**	**57.7 (↓0.6)**
Retention Ratio=15%					
VidCom^2 [EMNLP'25]	**859.0 (↓82.0%)**	**3381.4 (↓53.6%)**	**15.8** (↓10.7%)	**0.92 (1.44×)**	54.2 (↓4.1)
MoPrune	859.2 (↓82.0%)	3457.7 (↓52.6%)	**15.8 (↓10.7%)**	**0.92 (1.44×)**	**57.7 (↓0.6)**

Table 2: **Efficiency comparisons on MVBench with LLaVA-OV-7B.** "LLM Generation Latency": time for LLM-only response generation; "Model Generation Latency": time for model to generate response; and "Throughput": number of MVBench samples processed per second.

VideoMME (Fu et al., 2025a), detailed in Appendix A.

Baselines. We compare MoPrune with a variety of untrained token pruning strategies, including FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2024a), PrundVid (Huang et al., 2025), DyCoke (Tao et al., 2025), VisionZip (Yang et al., 2025), FrameFusion (Fu et al., 2025b), and VidCom^2 (Liu et al., 2025b), detailed in Appendix C. We use an equal retention rate for fair comparison, defined as the average percentage of visual tokens processed across all layers of the LLM.

Implementation Details. We evaluated on two representative VideoLLMs: LLaVA-OneVision-7B (Li et al., 2024a) and LLaVA-Video-7B (Zhang et al., 2024b). Consistent with the official settings, we set the input frame rate to 32 for LLaVA-OneVision and 64 for LLaVA-Video. For PruneVid, which compresses visual tokens and KV cache, we evaluated only its token compression strategy. All experiments were performed on an NVIDIA A40 GPU.

4.2 Main Results

Performance Comparisons. Table 1 compares MoPrune with several state-of-the-art token pruning methods on various benchmarks. Experimental results show that MoPrune achieves best performance across all configurations, significantly outperforming other token pruning methods. Specifically, for LLaVA-OneVision-7B, MoPrune, with a 25% retention rate, outperforms the dense model baseline, reaching 100.5% of the original. This improvement is attributed to appropriate pruning that reduces video noise while minimizing redundant information and better focusing on visual key elements. Furthermore, it maintains high performance even with decreasing retention rates, reaching 99% of the original at a 15% retention rate, 4.2% higher than the baseline method VidCom^2. This demonstrates the effectiveness of scene-guided motion-aware token pruning, overcoming the shortcomings of relying solely on static uniqueness, and showing significant effects on both long and short videos. To further evaluate the cross-model robustness of MoPrune, the same retention rate test was performed on LLaVA-Video, and it still maintains its superiority over other competing methods, achieving 95.7% and 93.9% of the original performance at 25% and 15% retention rates, respectively.

Metrics	MLVU	VideoMME				Avg.
		Overall	Short	Medium	Long	
Vanilla	63.0	58.5	70.2	56.6	48.8	100.0
w/o scene	61.4	58.1	**70.1**	55.7	48.7	98.4
uniform γ_t	61.7	58.3	69.4	56.6	48.9	98.8
w/o $m_{t,n}$	61.0	56.2	66.6	53.7	48.4	96.4
w/o $u_{t,n}$	62.2	57.1	66.7	56.7	47.9	98.2
w/o $u_{t,n}^{\text{frame}}$	61.9	57.9	68.9	56.4	48.4	98.6
w/o $u_{t,n}^{\text{scene}}$	61.9	58.3	69.4	**57.0**	48.4	99.0
all	**62.3**	**58.5**	**70.1**	56.3	**49.0**	**99.4**

Table 3: Effects of scene segmentation, adaptive frame adjustment and different token score.

Efficiency Comparisons. Table 2 compares the real-world inference efficiency and GPU memory usage of different token pruning methods on MVBench. The Flash Attention 2 operator was used for acceleration, and the comparison was conducted fairly on a single NVIDIA A40 GPU. MoPrune only slightly increases LLM generation latency and model generation latency compared to VidCom^2, while maintaining the same GPU memory usage. However, it achieves a 3.5 performance improvement at a 15% retention rate and

Method	MLVU	VideoMME				Avg.	Time
		Overall	Short	Medium	Long		
Vanilla	63.0	58.5	70.2	56.6	48.8	100.0	1.0
DOF	61.4	57.1	67.4	54.8	49.0	97.5	↓12.7%
SOF	61.6	56.8	68.7	54.0	47.8	97.4	↓45.4%
MB	61.8	57.1	68.1	54.6	48.6	97.9	↓48.0%
FD	**62.3**	**58.5**	**70.1**	**56.3**	**49.0**	**99.4**	↓52.6%

Table 4: Ablations on Motion Estimation. DOF: dense optical flow (Farnebäck, downsampled to patch grid). SOF: sparse optical flow (PyrLK on regular grid). MB: Macroblock. FD: feature difference (L2). Time: Model Generation Latency (s)

τ	MLVU	VideoMME				Avg.
		Overall	Short	Medium	Long	
Vanilla	63.0	58.5	70.2	56.6	48.8	100.0
0.6	61.9	58.0	69.8	55.8	48.6	98.7
0.7	61.9	58.3	69.9	**56.4**	48.4	99.0
0.8	**62.3**	**58.5**	**70.1**	56.3	**49.0**	**99.4**
0.9	61.8	57.7	68.0	56.1	48.9	98.4

Table 5: Ablations on different threshold τ for scene segmentation.

also shows improvement at a 25% retention rate, achieving a better balance between efficiency and performance compared to all other methods. Furthermore, like DyCoke and VidCom^2, our method is compatible with Flash Attention and does not introduce additional memory overhead.

4.3 Ablation Studies

We conducted ablation studies and analyses on LLaVA-OV-7B, using a 15% retention rate, to verify the rationality of the MoPrune design and the selection of parameters.

Effects of Scene Segmentation and Adaptive Frame Adjustment. Table 3 illustrates the effects of scene segmentation and frame adaptive adjustment. Scene segmentation benefits long videos more, as their numerous and complex scenes make per-scene motion amplitude calculation more effective. Furthermore, the large boundaries of scene transitions can easily cause interference, making it unsuitable to consider motion scores. Adaptive frame adjustment, on the other hand, can emphasize more unique frames within a scene, thereby maximizing visual content. As shown in the table, it is effective for both long and short videos.

Effects of Different Token Score. Table 3 shows the impact of different components of the token score. By comparing the performance under the conditions of removing the motion score $m_{t,n}$, uniqueness score $u_{t,n}$, scene-level uniqueness score $u_{t,n}^{\text{scene}}$, and frame-level uniqueness score $u_{t,n}^{\text{frame}}$, it demonstrates the effectiveness of the two-level uniqueness score and reflects the complementarity of motion score and uniqueness score. Their combination further achieves the overall goal of preserving visually unique or highly motion-oriented tokens.

Ablations on Motion Estimation. Table 4 compares the performance of different motion estimation methods. It can be seen that the simple feature difference not only has the best performance but also the best acceleration effect. The performance difference between dense optical flow and sparse optical flow is not significant because each token only needs one estimate, but dense optical flow is significantly slower. In contrast, the macroblock-based method is inferior to feature difference in both accuracy and efficiency.

Ablations on Different Parameter. We conducted ablation experiments with different scene segmentation thresholds τ, and the specific results are shown in Table 5. Optimal performance was achieved at $\tau = 0.8$. A threshold that is too large can lead to overly fragmented scene segments, resulting in many excessively small scene fragments that reduce the effectiveness of motion scores and lead to nearly uniform frame retention. A threshold that is too small can easily lead to semantically incoherent scenes, interfering with the estimation of scene representations.

5 Conclusion

In this work, we propose MoPrune, a novel training-free video token pruning framework for accelerating VideoLLMs. MoPrune leverages scene segmentation to maintain temporal and motion consistency and determines frame retention rates based on the uniqueness of frames within a scene. At the frame level, it performs motion-aware and uniqueness-driven token selection, prioritizing tokens with larger motion amplitudes while preserving unique visual content, significantly reducing redundant tokens. Extensive evaluations on multiple VideoLLMs and benchmarks demonstrate that MoPrune consistently achieves a superior efficiency–performance trade-offs and is fully compatible with hardware-efficient techniques such as Flash Attention, enabling seamless integration into existing VideoLLM pipelines.

6 Limitations

We propose a plug-and-play pruning framework for VideoLLMs that leverages scene-guided motion perception for efficient inference, but challenges remain. We employed feature differences as a simple motion estimation method, achieving a remarkable balance between efficiency and performance. We also explored optical flow estimation, but it struggled to capture complex video dynamics. Future work could explore refined motion estimation methods to achieve fine-grained, sensitive video pruning. Furthermore, it is necessary to expand its practical applications to adapt to different model configurations.

References

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

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, and 1 others. 2024b. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025a. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2025b. 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.

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

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024b. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2025a. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024c. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. 2025b. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, pages 1–19.

Ruyang Liu, Shangkun Sun, Haoran Tang, Wei Gao, and Ge Li. 2025a. Flow4agent: Long-form video understanding via motion prior from optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23817–23827.

Xuyang Liu, Yiyu Wang, Junpeng Ma, and Linfeng Zhang. 2025b. Video compression commander: Plug-and-play inference acceleration for video large language models. *arXiv preprint arXiv:2505.14454*.

Xuyang Liu, Zichen Wen, Shaobo Wang, Junjie Chen, Zhishan Tao, Yubo Wang, Tailai Chen, Xiangqi Jin, Chang Zou, Yiyu Wang, and 1 others. 2025c. Shifting ai efficiency from model-centric to data-centric compression. *arXiv preprint arXiv:2505.19147*.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, and 1 others.

2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232.

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

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and 1 others. 2024. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*.

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

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. 2024b. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, and 1 others. 2025a. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2025b. LMMs-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916.

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

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024a. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

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

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. 2025. Efficient motion-aware video mllm. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24159–24168.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2024. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*.

Appendix

A Benchmark Details

We evaluate MoPrune on multiple benchmarks, and their details are as follows:

- **MVBench** (Li et al., 2024c) formulates 20 video understanding tasks that require deep comprehension of temporal dimensions, beyond single-frame analysis.
- **LongVideoBench** (Wu et al., 2024) focuses on long-context video understanding with 3,763 videos up to one hour long. It includes 6,678 multiple-choice questions across 17 categories, emphasizing temporal information retrieval and analysis.
- **MLVU** (Zhou et al., 2024) features videos ranging from 3 minutes to 2 hours, encompassing 9 evaluation tasks including topic reasoning, anomaly recognition, video summarization, and plot question-answering.
- **VideoMME** (Fu et al., 2025a) comprises 900 videos and 2,700 multiple-choice questions across six domains, with durations from 11 seconds to 1 hour, categorized into short, medium, and long subsets.

B Model Details

We evaluated MoPrune on two VideoLLMs, and their details are as follows:

- **LLaVA-OneVision** (Li et al., 2024a) unifies single-image, multi-image, and video tasks in a single LLaVA-OneVision model. It represents videos as long visual token sequences in the same "interleaved" format used for images, enabling smooth task transfer from images to videos and facilitating strong zero-shot video understanding capabilities.
- **LLaVA-Video** (Zhang et al., 2024b) builds upon the single-image stage checkpoint of LLaVA-OneVision. It is fine-tuned on a large synthetic video-instruction dataset (LLaVA-Video-178K), covering detailed captioning, open-ended QA, and multiple-choice QA. By employing the SigLIP visual encoder and Qwen2 as the LLM, LLaVA-Video achieves robust video comprehension across various benchmarks.

C Baseline Details

We compared several advanced VLMs pruning strategies, including some specific ones for VideoLLMs, and their details are as follows:

- **FastV** (Chen et al., 2024a) performs one-time token pruning as an intra-LLM compression method, utilizing attention weights associated with the output token after a selected LLM layer. However, its explicit dependence on attention weights makes it incompatible with Flash Attention (Dao et al., 2022) in LLM.
- **SparseVLM** (Zhang et al., 2024a) functions as an intra-LLM compression method, ranking token importance using text-visual attention maps and pruning via pre-selected text prompts to mitigate attention noise. Similar to FastV, SparseVLM is also incompatible with Flash Attention (Dao et al., 2022) in LLM.
- **DyCoke** (Tao et al., 2025) is a two-stage VideoLLM-specific approach that first prunes temporally similar tokens and then compresses less-attended visual tokens in the KV cache using LLM attention weights. Its reliance on partitioning frame sets and similarity-based compression limits aggressive one-shot token reduction. Although its token pruning stage is compatible with Flash Attention (Dao et al., 2022), the KV cache compression depends on explicit attention weights, making it incompatible with efficient attention operators.
- **VisionZip** (Yang et al., 2025) selects important tokens using the average attention each token receives from all others in the sequence and then merges the remaining ones.
- **PruneVid** (Huang et al., 2025) clusters frames into segments and classifies tokens within each segment as either static or dynamic, applying different pruning strategies accordingly. While the original PruneVid compresses both visual tokens and KV Cache, to ensure a fair comparison, we only implement the pre-LLM pruning.
- **FrameFusion** (Fu et al., 2025b) is motivated by two key observations. First, spatially corresponding visual tokens between adjacent

frames exhibit significantly higher cosine similarities than other token pairs. Second, although the similarity between highly similar tokens decreases at deeper layers, the relative similarity rankings of these tokens remain stable. Inspired by this strong consistency, FrameFusion adopts a two-stage approach: it first merges tokens across frames and subsequently selects important ones.

- **VidCom**^{2} (Liu et al., 2025b) reveals two critical issues: ignoring frame-wise visual diversity and facing implementation constraints that hinder compatibility with modern architectures and efficient operators. VidCom^2 addresses these issues by quantifying the uniqueness of each frame to adaptively adjust compression strength across frames, and further measuring token uniqueness using both video-level and frame-level uniqueness scores, thereby effectively preserving the most distinctive tokens within each frame and across the entire video.

D AI Assistants in Research and Writing

Yes, we utilized AI assistants in certain aspects of our research and writing process. Specifically, generative AI tools such as ChatGPT were used to assist in drafting portions of the Python code and parts of the appendix, as well as in polishing and refining sections of the manuscript. These tools were primarily employed to enhance clarity, improve grammatical structure, and achieve a more concise presentation of our ideas.

We emphasize that while AI-assisted tools supported parts of the writing and code generation process, all core research activities—including problem formulation, methodology design, experimental analysis, and interpretation of results—were conducted independently by the authors. The use of AI tools was limited to auxiliary tasks and did not affect the integrity, originality, or scientific contributions of the work. All final content was carefully reviewed and validated to ensure academic rigor and accuracy.