METok: Multi-Stage Event-based Token Compression for Efficient Long Video Understanding

Mengyue Wang[1,7] **Shuo Chen**[2,6,7] **Kristian Kersting**[3,4,5]
Volker Tresp[2,7] **Yunpu Ma**[2,7 †]
[1]Technical University of Munich [2] LMU Munich [3]DFKI SAINT [4]Hessian AI
[5]TU Darmstadt [6]Konrad Zuse School of Excellence in Reliable AI (relAI)
[7]Munich Center for Machine Learning (MCML)

Abstract

Recent advances in Vision Large Language Models (VLLMs) have significantly enhanced their ability to understand video content. Nonetheless, processing long videos remains challenging due to high computational demands and the redundancy present in the visual data. In this work, we propose **METok**, a training-free, **M**ulti-stage **E**vent-based **Tok**en compression framework designed to accelerate VLLMs' inference while preserving accuracy. METok progressively eliminates redundant visual tokens across three critical stages: (1) event-aware compression during vision encoding, (2) hierarchical token pruning in the prefilling stage based on semantic alignment and event importance, and (3) a decoding-stage KV Cache optimization that further reduces memory consumption. Our experiments on diverse video benchmarks demonstrate that METok achieves an optimal trade-off between efficiency and accuracy by dynamically selecting informative visual tokens. For instance, equipping LongVA-7B with METok realizes an 80.6% FLOPs reduction and 93.5% KV Cache memory savings, all while maintaining comparable or even superior accuracy. The code is available here.

1 Introduction

Vision Large Language Models (VLLMs) (Cheng et al., 2024; Wang et al., 2024; Lin et al., 2023; Li et al., 2024d; Liao et al., 2024a; Bi et al., 2025b) have recently achieved remarkable success in various tasks such as video question-answering, temporal reasoning, and grounding. However, extending these models to efficiently understand long videos remains a major challenge. Current VLLMs typically encode each frame into hundreds of tokens, leading to prohibitive computational and memory overhead as sequence lengths scale. Beyond the resource burden, this visual token proliferation also introduces semantic redundancy, where repetitive or low-salience tokens obscure truly informative visual cues. As the number of frames increases, critical visual information becomes diluted, ultimately degrading both model efficiency and performance.

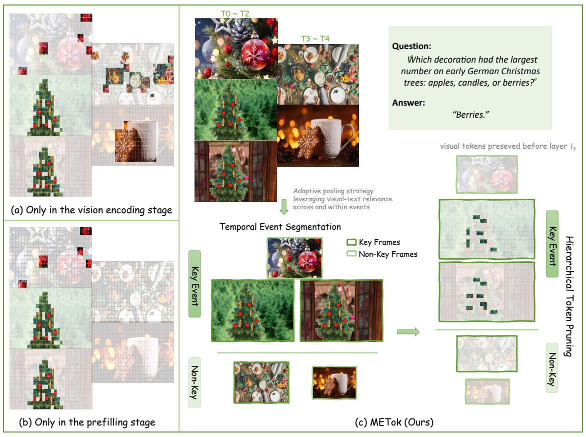

(a) Only in the vision encoding stage
(b) Only in the prefilling stage
T0 ~ T2
T3 ~ T4
Question:
"Which decoration had the largest number on early German Christmas trees: apples, candles, or berries?"
Answer:
"Berries."
visual tokens preseved before layer l_3
Adaptive pooling strategy leveraging visual-text relevance across and within events
Temporal Event Segmentation
Key Frames
Non-Key Frames
Key Event
Non-Key
Key Event
Non-Key
Hierarchical Token Pruning
(c) METok (Ours)

Figure 1: Comparison of visual token compression methods: (a) methods that only compress tokens during the vision encoding stage, such as VisionZip (Yang et al., 2024); (b) methods that drop tokens only in the prefilling stage like FastV (Chen et al., 2024); (c) METok (ours), which progressively removes irrelevant tokens in three stages (vision encoding, prefilling, and decoding).

To address these challenges, previous work (Bolya et al., 2022; Chen et al., 2024; Shang et al., 2024; Xu et al., 2024; Fu et al., 2024b; Ren et al., 2023) has explored various token compression strategies to optimize VLLM inference. However, most of these approaches operate on a single-stage level, and do not fully account for how the relevance of visual information evolves across different layers and stages of inference. Such ignorance may lead to overlooking cross-modal alignment or discarding features that may only become salient in deeper layers.

Consequently, an important question arises: *How can we effectively reduce redundant visual tokens while preserving informative spatiotemporal*

[†] Corresponding author: cognitive.yunpu@gmail.com

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18870–18884
November 4-9, 2025 ©2025 Association for Computational Linguistics

content throughout the entire inference process?

To answer this question, we analyze existing VLLMs across inference stages and derive two key insights: (1) The critical video information is task-dependent and temporally sparse, underscoring the need to preserve spatiotemporal relationships while retaining text-relevant visual tokens. (2) Visual tokens primarily contribute in shallow LLM layers, with their influence diminishing in deeper layers as attention shifts to text tokens.

Building upon these observations, we propose **METok**, a **M**ulti-stage **E**vent-based **Tok**en compression framework that integrates visual-text alignment with hierarchical token reduction, without requiring additional training. METok operates in three stages: During the vision encoding stage, it segments videos into temporally coherent events and identifies key visual tokens while heavily compressing non-key ones. In the prefilling stage, it performs layer-wise hierarchical token pruning guided by attention and event importance. Finally, METok discards visual tokens from the KV Cache starting at the pruning boundary used in prefilling. We validate METok across diverse models and benchmarks, demonstrating that it can reduce computational cost by up to 80.6% and KV cache memory by over 90%, all while maintaining or even improving accuracy, and requiring no additional training. These three stages jointly improve memory and computational efficiency without compromising model performance, making METok an effective solution for long video understanding in VLLMs. The main contributions are summarized as follows:

- We propose **METok**, a training-free, plug-and-play token compression framework for long video understanding, applicable across the entire VLLM inference pipeline.
- METok employs stage-specific compression strategies that preserve text-relevant and spatiotemporal information through semantic-aware token pruning.
- Extensive experiments demonstrate that METok substantially reduces computational demands while maintaining or improving performance over base VLLMs.

2 Related Work

2.1 Vision Large Language Models

The rapid growth of Large Language Models (LLMs) (Bai et al., 2023; Chiang et al., 2023; Touvron et al., 2023a) has further accelerated the development across a wide range of areas (Liao et al., 2024b; Zhang et al., 2025, 2024b; Bi et al., 2025a) and notably in VLLMs, where strong language backbones are extended with vision encoders. To extend these capabilities from images to videos, recent works have proposed various strategies for integrating temporal visual inputs into LLMs. LLaVA-OneVision(Li et al., 2024a) unifies image and video tasks within a single model that supports cross-modal transfer. LongVA(Zhang et al., 2024b) scales video comprehension by extrapolating LLM's context length, allowing inference over thousands of visual tokens. However, most VLLMs still process video frames independently, encoding each frame into its own set of tokens. While suitable for short clips, this approach becomes prohibitively expensive for long videos due to the quadratic complexity of self-attention.

2.2 Token Compression

Token compression techniques (Bolya et al., 2022; Zhong et al., 2024; Fu et al., 2024c; Yang et al., 2024; Wan et al., 2024; Liu et al., 2024; Shen et al., 2024; Tao et al., 2024; Xing et al., 2025) in VLLMs can be categorized by the inference stage at which they operate. For instance, AuroraCap (Chai et al., 2025) merges similar visual tokens within transformer layers, LVC (Wang et al., 2025) introduces a parameter-free query-attention video compression mechanism, St3 (Zhuang et al., 2025) prunes inattentive visual tokens progressively across LLM layers, VideoChat-Flash (Li et al., 2024c) segments video and applies hierarchical compression in two stages, and FastVid (Shen et al., 2025) adopts a density-based token pruning strategy to maintain essential information. Methods such as VisionZip (Yang et al., 2024) and DivPrune (Alvar et al., 2025) compress visual tokens early in the vision encoding stage. Prefilling-stage token compression approaches like FastV (Chen et al., 2024) remove redundant visual tokens after text interaction begins, typically by analyzing attention weights within LLM to drop those contributing the least to multimodal reasoning. While these techniques (Zhong et al., 2024; Tu et al., 2024; Xiao et al., 2023) bring efficiency gains, they often operate in isolation, overlooking the cumulative impact of redundancy across stages. In contrast, METok integrates token compression across all stages with spatiotemporal awareness, enabling efficient and effective processing of long video sequences.

3 Method

3.1 Preliminary

Task Formulation. Given a video V and a text input c, the goal is to generate a textual response y by maximizing the conditional probability

$$\max_{y} \; p\left(y \mid V, \, c\right) \tag{1}$$

Vision encoding stage. Each frame is divided into N patches $\{\mathbf{x}_i\}_{i=1}^{N}$, projected into embeddings with positional encoding and processed through L layers of Multi-Head Self-Attention (MHSA) (Vaswani et al., 2017):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d/H}}\right)\mathbf{V}, \tag{2}$$

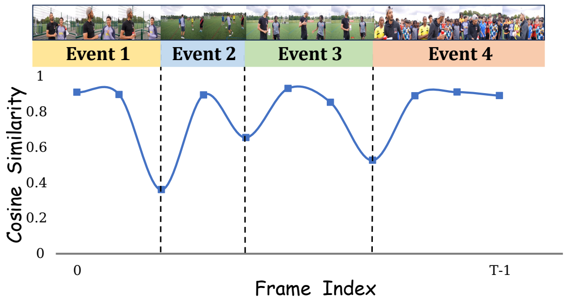


Figure 3: The visualization of temporal event segmentation based on the similarity of adjacent frames.

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d/H}$ are query, key, and value matrices distributed across H heads, and d denotes the feature dimension of each token.

Prefilling stage. The model encodes the concatenated input $\mathbf{X} = [\mathbf{X}_{\text{vis}}; \mathbf{X}_{\text{text}}] \in \mathbb{R}^{T \times d}$, and caches key-value pairs $\{\mathbf{K}_l, \mathbf{V}_l\}$ for each transformer layer l using projection matrices $\mathbf{W}_l^K$ and $\mathbf{W}_l^V$ as follows:

$$\mathbf{K}_l = \mathbf{X}\mathbf{W}_l^K, \quad \mathbf{V}_l = \mathbf{X}\mathbf{W}_l^V. \tag{3}$$

Decoding stage. The decoding process follows an autoregressive pattern, in which each token is predicted sequentially by referencing previously generated tokens and stored key-value (KV) pairs.

3.2 METok

METok follows a structured three-stage approach tailored for long-video inference to progressively remove redundant visual tokens, achieving an optimal balance between computational efficiency and preservation of spatiotemporal semantics throughout the entire pipeline, as shown in Figure 2.

3.2.1 Vision Encoding: Event-Aware Token Reduction

Temporal Event Segmentation. Long videos are naturally composed of multiple semantically distinct events that unfold over time. When all frames are treated equally, this temporal structure is ignored, resulting in inefficient processing and a diluted representation of meaningful visual content. To better reflect this structure, METok segments the video into meaningful temporal events based on content dynamics. Some existing methods like Chat-Univi (Jin et al., 2024) with DPC-KNN (Du et al., 2016), use clustering to capture event structure, but often ignore temporal continuity and produce fragmented segments. In contrast, METok detects meaningful transitions between frames to identify coherent event boundaries.

Given frame-level visual embeddings $V = \{v_i\}_{i=1}^{T} \in \mathbb{R}^{T \times N \times d}$, where T is the number of frames, N is the number of tokens per frame, and d is the embedding dimension, the cosine similarity between adjacent frame embeddings v_i and v_{i+1} is computed as

$$S_i = \frac{v_i \cdot v_{i+1}}{\|v_i\| \|v_{i+1}\|}, \quad i \in [0, T-1]. \tag{4}$$

As shown in Figure 3, a lower similarity score indicates a significant change in content, suggesting a potential event boundary. We select the $k-1$ lowest S_i scores as event boundaries, dividing the video into k events. Let E denote the set of events.

Key Visual-Text Sematic Identification. After event segmentation, METok estimates the text relevance of each frame to distinguish semantically important content from less informative visual information. we observed that many existing VLLMs, such as LLaVA-OneVision (Li et al., 2024a) and LongVA (Zhang et al., 2024b), employ CLIP (Radford et al., 2021) or SigLIP (Zhai et al., 2023) ViT-L as their vision encoders, which are pretrained for visual-text alignment. To leverage this, METok reintroduces the associated text encoder to compute the cross-modal similarity score $S_{v_i,t}$ between the visual embeddings $v_i \in \mathbb{R}^{N \times d}$ of each frame and the encoded text embedding $t \in \mathbb{R}^{1 \times d}$:

$$S_{v_i,t} = \frac{v_i \cdot t}{\|v_i\| \|t\|}, \quad i \in [0, T-1]. \tag{5}$$

We average these scores per event and rank all k events accordingly. The top $\lceil \alpha \cdot k \rceil$ events $(0 < \alpha < 1)$ are designated as key events E_{key} while the remaining are considered non-key events $E_{\text{non-key}}$.

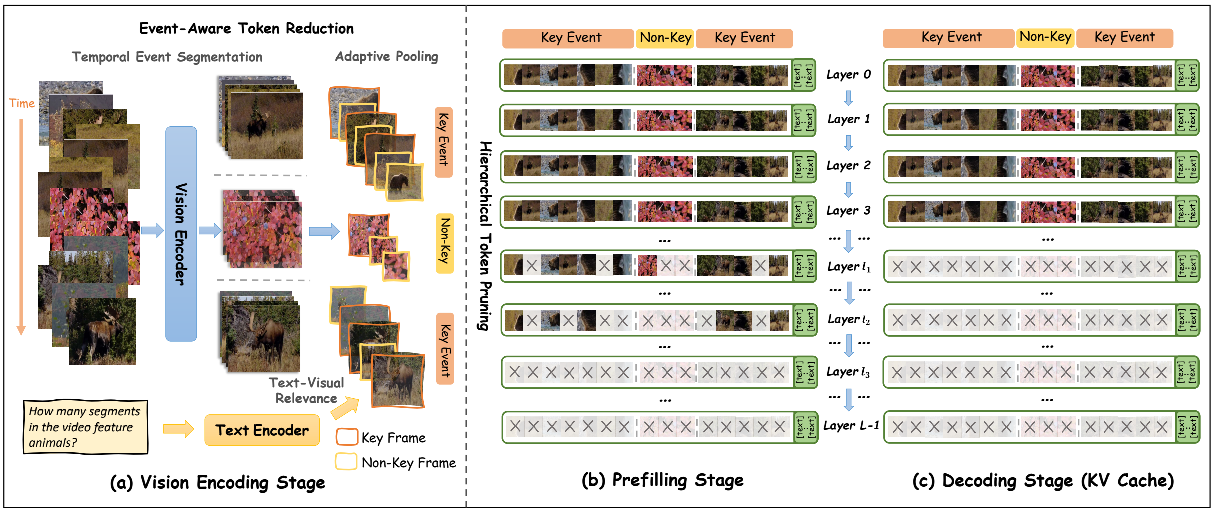


Figure 2: The architecture of METok. In the vision encoding stage, METok first segments video into events and compresses key events and frames based on visual-text relevance. Then, it hierarchically prunes redundant visual tokens using attention and event importance in the prefilling stage. Finally, the prefill-driven KV Cache Optimization further removes visual tokens starting at layer l_1.

Within each event, whether key or non-key, METok further selects the top β proportion of frames ($0 < \beta < 1$) that convey richer textual semantics as key frames, treating all remaining frames as non-key frames.

Adaptive Pooling Strategy. METok applies an adaptive pooling strategy to refine compression while preserving temporal relationships. It applies *different pooling granularities to key and non-key frames of key and non-key events*. Specifically, we define different pooling strides s_1 and s_2 for key and non-key frames in each event of E_{key}. To maintain key events at a higher resolution while aggressively downsampling non-key events, we further adjust the pooling strides s_1 and s_2 by a factor of $\frac{1}{\alpha} > 1$ for events in $E_{\text{non-key}}$, ensuring that non-key events undergo higher pooling. This approach ensures the retention of critical event and frame details while removing superfluous content, thereby optimizing both event-level and frame-level token compression for long video understanding.

3.2.2 Prefilling: Hierarchical Token Pruning

Although the vision encoding stage filters out many redundant tokens, some visually marginal content still remains. If not further processed, these tokens persist through the model's inference, occupying memory and compute without meaningfully contributing to the final output. To address this, METok introduces a hierarchical token pruning strategy in the prefilling stage, ensuring a progressive refinement of visual token selection.

The design builds on the observation that early layers in LLMs primarily extract low-level visual features while deeper layers gradually emphasize semantically relevant elements. Leveraging this shift, METok prunes visual tokens in a layer-wise manner guided by visual-text attention scores. At each layer, only the top-ranked tokens—those most semantically aligned with the text input—are retained, ensuring that models allocate resources to the informative content throughout inference.

Concretely, METok retains a higher proportion of tokens for key events and prunes more aggressively for non-key events, with the relative pruning rates modulated by the key event ratio α (see Eq. 7) The pruning process unfolds across three compression levels defined by the layer boundaries $L_H = [l_1, l_2, l_3]$. Token retention ratios are expressed relative to the number of visual tokens originally passed into the LLM. The visual token retention ratio for key events, $R_{\text{key}}(l)$, and non-key events, $R_{\text{non-key}}(l)$, across different layers l are defined as:

$$R_{key}(l) = \begin{cases} 1, & \text{if } l < l_1 \\ r, & \text{if } l_1 \leq l < l_2 \\ r^2, & \text{if } l_2 \leq l < l_3 \\ 0, & \text{if } l > l_3 \end{cases} \tag{6}$$

$$R_{non\_key}(l) = \begin{cases} 1, & \text{if } l < l_1 \\ \alpha \cdot r, & \text{if } l_1 \leq l < l_2 \\ 0, & \text{if } l > l_2 \end{cases} \tag{7}$$

where r is a pruning factor ($0 < r < 1$), and α is the proportion of key events.

This progressive pruning process ensures that early layers preserve necessary visual information, intermediate layers refine essential semantic information, and later layers retain only the most relevant text-aligned features, allowing for efficient and effective processing of long videos.

3.3 Decoding: Prefilling-driven Optimization

Despite the effectiveness of prior token pruning in reducing visual redundancy, optimizing memory efficiency throughout the entire inference process-particularly during decoding-remains challenging. Most VLLMs still maintain all visual tokens in the KV Cache across all layers, resulting in substantial memory overhead.

To better understand this inefficiency, we analyze the attention distribution between visual and text tokens across decoding layers in various VLLMs like VILA-1.5 (Lin et al., 2024), as shown in 4. Notably, attention to visual tokens drops sharply after the early decoding layers, suggesting that these tokens contribute little to response generation in deeper layers. In contrast, text tokens consistently receive high attention across all layers, underscoring their continued relevance. Motivated by this, METok refines the KV Cache by aligning it with the hierarchical pruning strategy from the prefilling stage. Specifically, while both visual and text tokens are cached in the early decoding layers, METok removes entire visual tokens from the KV Cache starting at layer l_1—the same boundary where pruning begins during prefilling. This selective preservation substantially reduces memory consumption while maintaining essential multimodal context for accurate generation.

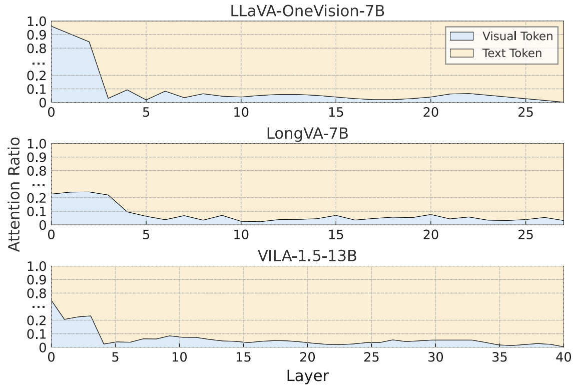


Figure 4: Layer-wise attention ratio between visual and text tokens during decoding in various VLLMs.

4 Experiments

Benchmarks. We conduct evaluation on widely used video understanding benchmarks, including Egoschema (Mangalam et al., 2023), MVBench (Li et al., 2024b), MLVU (Zhou et al., 2024), and VideoMME (Fu et al., 2024a). Notably, VideoMME (1 min $\sim$ 1 hr) and MLVU (3 mins $\sim$ 2 hrs) target long video scenarios, making them suitable for assessing long-context comprehension. We use `lmms-eval` (Zhang et al., 2024a) as our primary evaluation framework to uniformly assess performance across these diverse benchmarks. The details of these benchmarks are shown in AppendixA.

4.1 Implementation Details

We evaluate METok using various open-source VLLMs spanning small to large scales, including LLaVA-OneVision-(0.5B, 7B) (Li et al., 2024a), LongVA-7B (Zhang et al., 2024b) and VILA-1.5-13B (Lin et al., 2024), respectively sampling 32, 128, and 16 frames. LLaVA-OneVision and VILA-1.5 adopt a SigLIP-pretrained ViT-L as their vision tower, while LongVA employs a CLIP-pretrained ViT-L. To assess computational efficiency, we measure the total FLOPs of both the prefilling and decoding stages using calflops (Ye, 2023). Prefill Time refers to the latency required to generate the first token. More implementation details are included in the Appendix C.

4.2 Main Results

Comparison with base models. Table 1 presents the performance of METok on LLaVA-OneVision-(0.5B, 7B), LongVA-7B and VILA-1.5-13B across representative benchmarks. METok consistently achieves substantial reductions in FLOPs while maintaining or improving accuracy, demonstrating a trade-off between efficiency and performance. In particular, METok shows clear advantages in long-context scenarios. On the long subset of VideoMME, it consistently surpasses all baselines, indicating its ability to retain essential temporal and semantic visual tokens under extended durations. The results also highlight METok's scalability across model sizes, from LLaVA-OneVision-0.5B to VILA-1.5-13B, achieving up to 80.6% FLOPs reduction without degrading performance. Notably, for each base model, we use a single set of hyperparameters across all benchmarks without task-specific tuning. METok consistently delivers strong performance under this unified setting,

Model	#Frames	FLOPs (TB)	MVBench	Egoschema	MLVU	VideoMME (*w/o sub.*)		VideoMME (*w sub.*)	
						Overall	Long	Overall	Long
Duration			16 sec	179.8 sec	3∼120 min	1∼60 min	30∼60 min	1∼60 min	30∼60 min
LLaVA-OneVision-0.5B (Li et al., 2024a)	32	3.9	45.5	26.8	50.3	44.0	35.4	43.5	38.7
FastV (Chen et al., 2024)	32	2.2	**45.0**	26.5	50.8	42.4	35.8	46.4	39.4
VisionZip (Yang et al., 2024)	32	1.7	41.4	**27.3**	44.4	37.6	35.6	41.9	36.1
DivPrune (Alvar et al., 2025)	32	**1.2**	44.2	26.1	37.4	42.6	36.0	46.4	39.5
METok (Ours)	32	1.3	44.7	25.8	**52.1**	**42.7**	**36.4**	**47.0**	**39.7**
LLaVA-OneVision-7B (Li et al., 2024a)	32	71.4	56.7	60.1	64.7	58.2	48.5	61.2	51.4
FastV (Chen et al., 2024)	32	39.0	57.2	**60.1**	64.0	58.2	48.4	**61.8**	51.6
VisionZip (Yang et al., 2024)	32	27.7	49.4	56.4	55.0	49.6	44.6	55.2	48.4
DivPrune (Alvar et al., 2025)	32	20.9	55.9	58.2	60.8	54.4	46.6	58.9	50.8
METok (Ours)	32	**19.8**	**57.3**	59.8	**64.7**	**58.4**	**49.8**	61.7	**52.8**
LongVA-7B (Zhang et al., 2024b)	128	241.6	50.8	43.5	59.0	52.6	46.2	54.3	47.6
FastV (Chen et al., 2024)	128	133.6	**50.9**	44.2	59.2	**52.9**	45.6	55.7	47.0
VisionZip (Yang et al., 2024)	128	107.9	46.3	36.2	52.0	45.4	37.9	51.4	41.0
DivPrune (Alvar et al., 2025)	128	50.1	50.6	43.9	58.7	52.4	45.9	55.8	47.4
METok (Ours)	128	**46.8**	50.7	**44.4**	**60.4**	52.4	**46.6**	**56.0**	**47.7**
VILA-1.5-13B (Lin et al., 2024)	16	84.1	50.9	50.4	50.4	49.5	42.4	53.3	46.7
FastV (Chen et al., 2024)	16	49.8	50.4	50.0	49.5	48.4	41.6	53.1	45.9
VisionZip (Yang et al., 2024)	16	29.1	45.8	47.6	47.1	43.8	39.4	49.4	44.9
DivPrune (Alvar et al., 2025)	16	20.3	**50.7**	50.4	50.3	48.9	41.8	53.5	46.3
METok (Ours)	16	**19.6**	50.5	**50.5**	**50.5**	**49.5**	**42.7**	**53.9**	**48.1**

Table 1: Performance and efficiency comparison across different methods and benchmarks. The best result of token compression methods is **bolded** and the second best is underlined.

Method	FLOPs (TB)	Prefill Time (ms)	KV Cache (MB)	MLVU
LongVA-7B	241.6	1535.9	1012.3	59.0
w/FastV	133.6	1004.5	535.1	59.2
w/Visionzip	107.9	629.0	452.2	52.0
w/DivPrune	50.1	417.7	240.6	58.7
w/METok (Ours)	**46.8**	**378.6**	**65.0**	**60.5**
LLaVA-OneVision-7B	71.4	400.4	299.8	64.7
w/FastV	39.0	231.3	162.7	64.0
w/Visionzip	27.7	155.6	116.2	55.0
w/DivPrune	20.9	137.5	65.8	60.8
w/METok (Ours)	**19.8**	**131.8**	**22.3**	**64.7**

Table 2: Efficiency comparison of FLOPs, Prefill Time and KV Cache Memory for token compression methods.

demonstrating robustness to hyperparameter variation.

Comparison with baseline methods. To further highlight METok's effectiveness, we compare it against other training-free token compression baselines such as FastV (Chen et al., 2024), VisionZip (Yang et al., 2024) and DivPrune (Alvar et al., 2025) (Table 1). While FastV can reduce some computational overhead, relying solely on prefilling-stage compression places a notable burden on the shallow layers of the LLM, limiting FLOPs reduction. For instance, FastV reduces FLOPs by only 44.7% on LongVA-7B, largely because its early removal of visual tokens can prematurely discard important information. VisionZip and DivPrune compress tokens during the vision encoding stage by merging or selecting based on redundancy. While they achieve higher FLOPs reduction (e.g., 63.7%), their one-stage design often removes fine-grained details essential for temporal reasoning, leading to accuracy drops across tasks. In contrast, METok distributes compression across all three inference stages, preserving visual-text alignment and progressively pruning irrelevant tokens. This design enables it to maintain key visual-text interactions and systematically discards irrelevant tokens with minimal accuracy degradation across evaluated benchmarks.

Efficiency Analysis. METok offers substantial efficiency and memory savings while preserving strong video understanding performance. We compare FLOPs, prefill time, and KV Cache memory usage against vanilla LongVA-7B and LLaVA-OneVision-7B, as well as the single-stage methods FastV and VisionZip. As shown in Table 2, METok consistently reduces FLOPs and prefill time with minimal or zero accuracy loss. For instance, with LLaVA-OneVision-7B, METok lowers FLOPs by 72.3%, prefill time by 67.1%, and KV Cache memory by 92.6%, yet still matches the original MLVU accuracy. In contrast, one-stage approaches that either prune tokens prematurely (VisionZip and DivPrune) or rely solely on shallow cross-modal attention (FastV) often experience larger performance drops at comparable compression levels. By progressively pruning in the vision encoder, prefilling, and decoding stages, METok more accurately discards low-impact tokens while retaining those

Method	FLOPs (TB)	KV Cache (MB)	MLVU
LLaVA-OneVision-7B	71.4	299.2	64.7
w/ Vision Encoding Event-Aware Token Reduction	41.1	171.8 (↓ 42.6%)	**64.8**
w/ Prefilling Hierarchical Token Pruning	19.8	82.8 (↓ 72.3%)	64.7
w/ Decoding Prefilling-driven Optimization	**19.8**	**22.3** (↓ 92.5%)	64.7

Table 3: Ablation study of specific strategy at vision encoding, prefilling, and decoding stage, respectively.

essential for video understanding.

4.3 Ablation Study

Multi-Stage Strategy. We conduct an ablation experiment to assess the contribution of each stage. As shown in Table 3, every stage reduces computational overhead while preserving or even slightly improving performance. In the vision encoding stage, our event-aware token compression strategy segments videos into meaningful events and adaptively retains visual tokens based on semantic relevance. This step alone cuts FLOPs and KV Cache usage by 42.6% and slightly boosts accuracy, showing that early-stage filtering removes redundancy while keeping key information.

In the prefilling stage, we introduce a hierarchical token pruning mechanism guided by text-visual attention scores and event importance, dynamically dropping low-relevance visual tokens. This approach further reduces FLOPs and KV Cache by 72.3% without sacrificing accuracy. Lastly, in the decoding stage, METok refines KV Cache retention by discarding tokens from shallower layers that no longer contribute to final outputs, resulting in an additional 92.5% cut in KV Cache usage. These tailored strategies at each stage precisely eliminate invalid visual tokens while retaining content critical for downstream tasks.

Temporal Event Segmentation. We also compare various event segmentation strategies on LLaVA-OneVision-7B. As shown in Figure 5, uniform segmentation lowers FLOPs to 27.9% of the baseline but noticeably hurts performance, while random segmentation further reduces FLOPs to 21.1% yet also degrades accuracy. In contrast, our temporal segmentation strategy strikes the optimal balance by reducing FLOPs to 27.7% while outperforming the other methods on both MLVU and VideoMME benchmarks.

Key Visual-Text Semantic Identification. We further investigate the impact of key visual-text semantic identification at the vision encoding stage. As shown in Table 4, randomly designating key events or frames reduces FLOPs but results in unstable performance; choosing both events and frames at random yields even lower costs but significantly degrades accuracy. These drops highlight the importance of *structured, semantic-aware selection*. In contrast, METok identifies semantically key events and frames, retaining only the most relevant tokens.

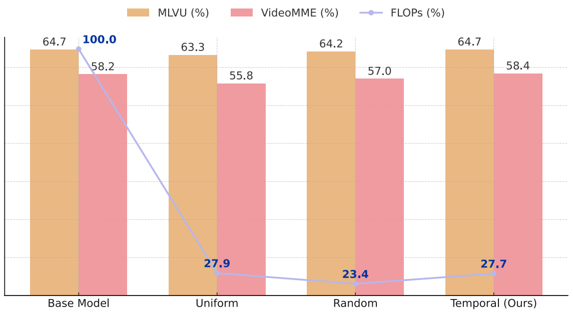


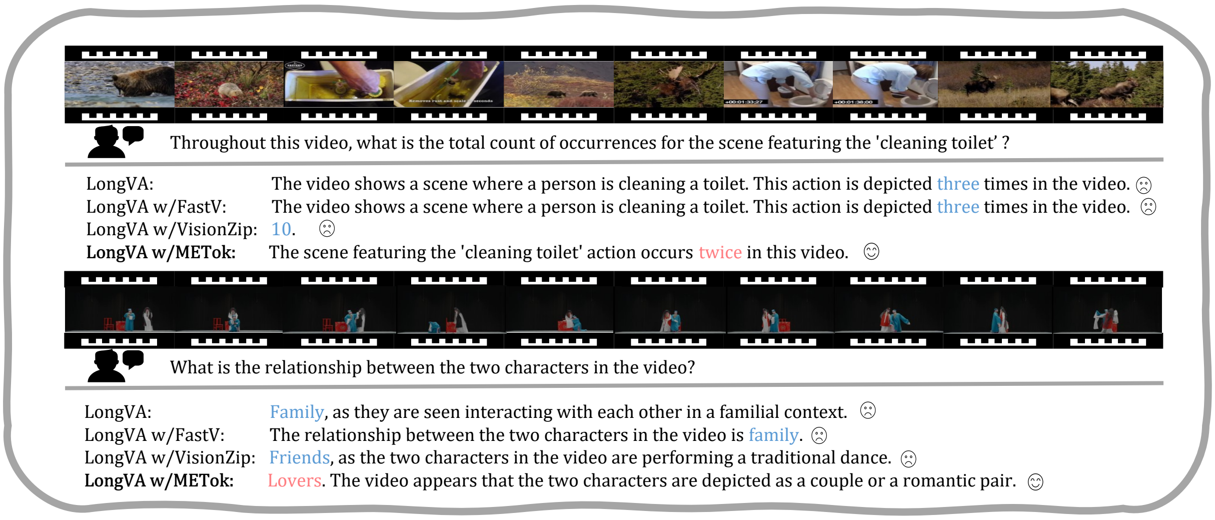
Figure 5: Comparison of FLOPs and accuracy (on MLVU and VideoMME) under different event segmentation strategies using LLaVA-OneVision-7B.

Method	FLOPs	MLVU	VideoMME
Base Model	100%	64.7	58.2
Random key events & frames	**21.1%**	60.9	54.6
Random key frames	27.8%	63.4	57.5
Random key events	22.3%	61.2	54.7
METok	27.7%	**64.7**	**58.4**

Table 4: Ablation of different key visual-text semantic identification strategy on LLaVA-OneVision-7B.

Adaptive Pooling Strategy. We ablate the adaptive pooling strategy by varying the pooling strides(s_1, s_2), with the key event ratio $\alpha = 0.5$ fixed. As shown in Figure 7, smaller strides retain more visual detail and improve performance, while larger strides result in lower FLOPs at the cost of degraded accuracy. The consistent trade-off across settings confirms that METok's adaptive pooling strategy effectively balances efficiency and semantic preservation, validating the overall effectiveness of the proposed design. Notably, even under highly aggressive pooling settings that reduce FLOPs to less than 10% of the base model, METok retains reasonable performance, highlighting its stability to hyperparameter choices.

Hierarchical Token Pruning. An ablation study on hierarchical token pruning with LLaVA-OneVision-7B (Table 5) demonstrates the importance of aligning pruning with the LLM's evolv-



Throughout this video, what is the total count of occurrences for the scene featuring the 'cleaning toilet' ?

LongVA: The video shows a scene where a person is cleaning a toilet. This action is depicted three times in the video. ☹
LongVA w/FastV: The video shows a scene where a person is cleaning a toilet. This action is depicted three times in the video. ☹
LongVA w/VisionZip: 10. ☹
LongVA w/METok: The scene featuring the 'cleaning toilet' action occurs twice in this video. ☺

What is the relationship between the two characters in the video?

LongVA: Family, as they are seen interacting with each other in a familial context. ☹
LongVA w/FastV: The relationship between the two characters in the video is family. ☹
LongVA w/VisionZip: Friends, as the two characters in the video are performing a traditional dance. ☹
LongVA w/METok: Lovers. The video appears that the two characters are depicted as a couple or a romantic pair. ☺

Figure 6: Qualitative results of METok compared to FastV and VisionZip with LongVA-7B on examples from VideoMME and MLVU benchmarks. Ground truth answers are referenced from the original benchmark annotations.

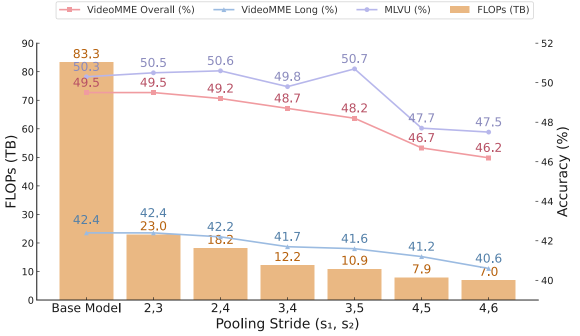


Figure 7: Ablation of adaptive pooling strategy on VILA-1.5-13B. Base Model refers to the uniform pooling setting with stride 2 for all frames.

ing semantic focus. Shallow pruning, which removes tokens aggressively in early layers, yields greater FLOPs savings but severely degrades accuracy by discarding critical low-level features too soon. Deep pruning, performed in later layers, better preserves essential information but reduces fewer FLOPs. In contrast, a balanced strategy achieves the best trade-off, cutting FLOPs to 27.7% while maintaining the baseline MLVU score of 64.7. By progressively pruning across all layers and dynamically adjusting token retention based on text-visual attention scores, METok prevents premature or excessive pruning, thus optimizing efficiency without sacrificing accuracy.

Selection of Scaling Factor r. Our ablation study on r shows a trade-off between efficiency and accuracy. Lower values like r = 0.3 or 0.4 overly prune tokens, while higher values like r = 0.7 preserve accuracy but provide limited computational savings. As shown in Figure 8, we select r = 0.55 as the optimal setting, as it achieves a strong balance, significantly reducing FLOPs while maintaining stable overall performance. Results on long videos further confirm METok's effectiveness, as compressed versions outperform the base model, suggesting excessive visual tokens may hinder VLMs in long-duration videos.

Pruning Strategy	$L_H = [l_1, l_2, l_3]$	**FLOPs**	**MLVU**
Base Model	without pruning	100%	64.7
Shallow Pruning	2,5,10	15.5%	53.0
	3,5,10	16.1%	53.1
	3,6,12	18.6%	55.1
Deep Pruning	15,23,26	46.5%	65.1
	17,23,27	48.5%	65.0
	19,24,28	50.7%	65.0
Balanced Pruning (Ours)	3,10,18	26.7%	64.5
	3,10,19	**27.7%**	**64.7**

Table 5: Ablation of different pruning strategies on LLaVA-OneVision-7B.

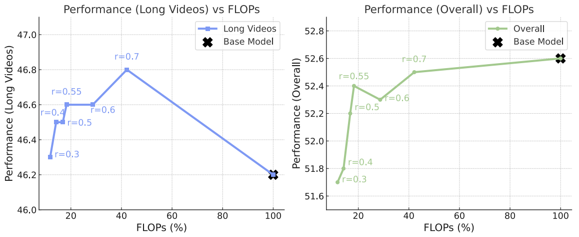


Figure 8: Ablation of different reduction scaling factor r with LongVA-7B on VideoMME, evaluating both overall and long video subset performance.

4.4 Qualitative Results

We present qualitative results in Figure 6 on more challenging multi-choice VideoQA tasks in VideoMME and MLVU, showcasing METok's effectiveness in long-video scenarios. Specifically, our approach accurately performs event segmentation and action counting, capturing key transitions while filtering redundant frames. On the object reasoning task, METok provides stronger multimodal alignment, enabling the model to discern emotional and contextual nuances in human interactions. These observations confirm METok's stable video-language understanding capabilities. See Appendix E for more qualitative examples.

5 Conclusion

In this paper, we introduced METok, a training-free multi-stage token compression framework that significantly enhances VLLM efficiency for long-video understanding. By integrating event-aware segmentation, hierarchical token pruning, and KV Cache optimization, METok substantially reduces computational costs while preserving accuracy. Our extensive experiments on LLaVA-OneVision-(0.5B, 7B), LongVA-7B and VILA-1.5-13B show that METok achieves up to 80.6% FLOPs reduction and 93.5% KV cache savings with comparable performance. METok generalizes well to different model sizes and architectures, while also improving memory consumption for the tested VLLMs. Overall, METok offers a unified and efficient solution that focuses on maintaining long-range temporal and semantic coherence, making it particularly well-suited for long-form video tasks.

Limitations

While recent progress has introduced vision large language models (VLLMs) exceeding 13B parameters, our current evaluation is limited to models up to 13B (e.g., VILA-1.5-13B), due to computational resource constraints. These larger models often require substantial GPU memory and infrastructure, which are beyond our reach at this stage. Nonetheless, we conduct comprehensive ablation studies across multiple open-source VLLMs and benchmarks, and the consistent improvements validate the general effectiveness of METok. We leave the evaluation on larger models to future work as resources become available.

Acknowledgements

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b273dd. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. Additionally, this work also benefited from the scientific support and HPC resources provided by The Hessian Center for Artificial Intelligence (hessian.AI), as well as the the Federal Ministry for Research, Technology and Space (BMFTR) project "XEI: Extremely Efficient Inference for Large Context Length" (XEI), project identification number 01IS24079B. This paper is also supported by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Research, Technology and Space.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. *arXiv preprint arXiv:2503.02175*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025a. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *ArXiv*, abs/2502.12119.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025b. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. *Preprint*, arXiv:2412.12359.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. 2025. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *Preprint*, arXiv:2410.03051.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. 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.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and 1 others. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Mingjing Du, Shifei Ding, and Hongjie Jia. 2016. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024a. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. 2024b. Lazyllm: Dynamic token pruning for efficient long context llm inference. *arXiv preprint arXiv:2407.14057*.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2024c. Framefusion: Combining similarity and importance for video token reduction on large visual language models. *arXiv preprint arXiv:2501.01986*.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. 2024. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710.

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*.

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

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, and 1 others. 2024c. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024d. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer.

Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2024a. Videoinsta: Zero-shot long video understanding via informative spatial-temporal reasoning with llms. *Preprint*, arXiv:2409.20365.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024b. Gentkg: Generative forecasting on temporal knowledge graph with large language models. *Preprint*, arXiv:2310.07793.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26689–26699.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. 2023. Testa: Temporal-spatial token aggregation for long-form video-language understanding. *arXiv preprint arXiv:2310.19060*.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. 2025. Fastvid: Dynamic density pruning for fast video large language models. *Preprint*, arXiv:2503.11187.

Leqi Shen, Tianxiang Hao, Tao He, Sicheng Zhao, Yifeng Zhang, Pengzhang Liu, Yongjun Bao, and Guiguang Ding. 2024. Tempme: Video temporal token merging for efficient text-video retrieval. *arXiv preprint arXiv:2409.01156*.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2024. Dycoke: Dynamic compression of tokens for fast video large language models. *arXiv preprint arXiv:2411.15024*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. 2024. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. *arXiv preprint arXiv:2410.23317*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. *arXiv preprint arXiv:2406.18139*.

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*.

Ziyi Wang, Haoran Wu, Yiming Rong, Deyang Jiang, Yixin Zhang, Yunlong Zhao, Shuang Xu, and Bo XU. 2025. Lvc: A lightweight compression framework for enhancing vlms in long video understanding. *Preprint*, arXiv:2504.06835.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *Preprint*, arXiv:2410.17247.

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

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*.

Xiaoju Ye. 2023. calflops: a flops and params calculate tool for neural networks in pytorch framework.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models. *Preprint*, arXiv:2407.12772.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024b. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. 2025. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23378–23386.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. 2024. Aim: Adaptive inference of multi-modal llms via token merging and pruning. *arXiv preprint arXiv:2412.03248*.

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*.

Jiedong Zhuang, Lu Lu, Ming Dai, Rui Hu, Jian Chen, Qiang Liu, and Haoji Hu. 2025. St3: Accelerating multimodal large language model by spatial-temporal visual token trimming. *Proceedings of the AAAI Conference on Artificial Intelligence*, (10):11049–11057.

A Datasets Details

We conduct comprehensive experiments comparing METok's performance with other training-free token compression methods on these video understanding benchmarks:

- MVBench (Li et al., 2024b): MVBench is a diagnostic benchmark for evaluating the temporal understanding abilities of multimodal large language models. It consists of 20 challenging tasks derived from existing static image benchmarks, transformed into video-based versions that require dynamic perception and reasoning. Each task uses a multiple-choice QA format based on curated YouTube video clips, with durations ranging from short to medium length. We conduct the zero-shot evaluation on the test set.
- EgoSchema (Mangalam et al., 2023): EgoSchema is a long-form video QA benchmark built on egocentric videos from the Ego4D dataset. It includes over 5,000 multiple-choice questions requiring reasoning over 3-minute video clips that capture real-world daily activities. Each question is paired with five answer options, and a subset of 500 questions includes human-annotated ground-truth labels. Following the standard evaluation setup, we report zero-shot performance on the test set.
- MLVU (Zhou et al., 2024): MLVU is a diagnostic benchmark designed to test different aspects of long video comprehension, including event identification, temporal reasoning, and narrative understanding. It features thousands of video clips ranging from 30 seconds to several minutes, with questions that span low-level recognition to high-level inference. We adopt the zero-shot setting and follow the official protocol on its dev set for evaluation.
- Video-MME (Fu et al., 2024a): Video-MME is a large-scale benchmark designed to comprehensively assess the video understanding capability of VLLMs. It contains 900 videos spanning 254 hours in total, with durations ranging from 11 seconds to over 60 minutes. The dataset covers 6 high-level categories and 30 subcategories, and includes manually annotated multiple-choice questions that require spatial-temporal reasoning and multi-event analysis. Our evaluation follows the official test set in a zero-shot manner.

B Pseudocode

We show the pseudo code of Event-aware Token Reduction during vision encoding stage in Algorithm 1.

Algorithm 1: METok - Vision Encoding

```
Input: Frame-level visual embedding
       ${v_1, \ldots, v_T}$; text embedding $t$
Parameter: Number of events $k$; key event
           ratio $\alpha$; key frame ratio $\beta$;
           pooling strides $s_1$, $s_2$
Output: Retained visual tokens $V_\mathrm{r}$
▷ Event temporal segmentation
$S_i \leftarrow \cos(v_i, v_{i+1}), \quad \forall i \in [1, T-1]$;
$E \leftarrow$ Split($V$) at $(k-1)$ lowest $S_i$
  positions;
▷ Key semantic identification
for $E_j \in E$ do
    for $v_i \in E_j$ do
        $S_{v_i,t} \leftarrow \cos(v_i, t)$;
    $r_j \leftarrow \max(S_{v_i,t})$;
    $F_j^{\text{key}} \leftarrow$ top-$\beta$ frames in $E_j$ by $S_{v_i,t}$;
    $F_j^{\text{non-key}} \leftarrow E_j \setminus F_j^{\text{key}}$;
$E_{\text{key}} \leftarrow$ top-$\lceil \alpha k \rceil$ events by $r_j$;
$E_{\text{non-key}} \leftarrow E \setminus E_{\text{key}}$;
▷ Adaptive pooling strategy
$V_\mathrm{r} \leftarrow []$;
for $E_j \in E$ do
    if $E_j \in E_{key}$ then
        for $v_i \in E_j$ do
            if $v_i \in F_j^{key}$ then
                $v_i^\mathrm{p} \leftarrow$ pool2d$(v_i, s_1)$;
            else
                $v_i^\mathrm{p} \leftarrow$ pool2d$(v_i, s_2)$;
            Append $v_i^\mathrm{p}$ to $V_\mathrm{r}$;
    else
        for $v_i \in E_j$ do
            if $v_i \in F_j^{key}$ then
                $v_i^\mathrm{p} \leftarrow$ pool2d$(v_i, s_1/\alpha)$;
            else
                $v_i^\mathrm{p} \leftarrow$ pool2d$(v_i, s_2/\alpha)$;
            Append $v_i^\mathrm{p}$ to $V_\mathrm{r}$;
return $V_r$
```

C Implementation Details

LLaVA-OneVision-(0.5B, 7B) and LongVA-7B are all built on top of Qwen. LLaVA-OneVision-7B and LongVA-7B use 28 transformer layers, while LLaVA-OneVision-0.5B has 24 layers. VILA-1.5-13B is based on LLaMA2 (Touvron et al., 2023b) and consists of 40 transformer layers.

We use a single set of hyperparameters for each base model across all benchmarks without task-specific tuning. For both LLaVA-OneVision-(0.5B, 7B) and LongVA-7B, we use $(s_1, s_2) = (2, 3)$, $L_H = [3, 10, 19]$, and $\alpha = 0.5$ as shared hyperparameters. The remaining parameters vary by model: for LLaVA-OneVision-(0.5B, 7B), we set $k = 5$, $\beta = 0.4$, and $r = 0.76$; for LongVA-7B, we set $k = 13$, $\beta = 0.45$, and $r = 0.55$. For VILA-1.5-13B, we use the setting of $(s_1, s_2) = (2, 3)$, $L_H = [13, 24, 34]$, $\alpha = 0.5$, $k = 3$, $\beta = 0.4$, and $r = 0.65$.

All experiments are conducted on NVIDIA A100 80 GPUs. To evaluate efficiency, we report FLOPs, Prefill Time, and KV Cache memory averaged across multiple video understanding benchmarks, including MLVU, VideoMME, MVBench, and EgoSchema.

D More Details of Experiments

D.1 Scalability to More Frames.

To assess METok's ability to handle longer videos under limited context, we conduct experiments on VILA-1.5-13B, which is based on LLaMA2 and supports a maximum context length of 4096 tokens. Under the default setting, VILA-1.5 accommodates only 16 frames without truncating visual or textual inputs. As shown in Table 6, by integrating METok, we have tested to scale the number of input frames from 16 to 64 within the same context budget.

Notably, as the number of frames increases, METok not only maintains performance but also brings consistent gains, especially on the MLVU and VideoMME benchmarks. For instance, at 64 frames, METok improves the MLVU score from 50.4 to 53.6, and VideoMME from 49.5 to 50.7. These results indicate that METok consistently achieves strong accuracy across different frame sampling densities and effectively compresses less informative tokens while preserving task-relevant visual semantics, allowing the model to benefit from richer temporal context.

Importantly, the tested frame counts (up to 64) do not yet represent the upper bound of METok's capability, and further improvements are expected with higher frame inputs. This demonstrates METok's strong scalability under strict context constraints and highlights its potential for long-form video understanding.

Below, we further compare different token reduction methods with VILA-1.5-13B under the 32-frame inputs. As shown in Table 7, METok achieves the lowest FLOPs while outperforming others in MVBench, MLVU, and VideoMME benchmarks. Notably, VILA-1.5-13B has a native maximum context length of 4096, which limits it to processing only 16 frames without any token reduction. Within this controlled setting, METok delivers better trade-off between accuracy and computation, indicating that its multi-stage, event-aware compression preserves task-relevant cues while minimizing redundant computation.

Method	#Frames	MVBench	MLVU	VideoMME
Base Model	16	50.9	50.4	49.5
w/METok	16	50.5	50.5	49.5
w/METok	32	50.9	52.8	49.9
w/METok	48	51.2	53.4	50.2
w/METok	64	50.7	53.6	50.7

Table 6: Evaluation of METok under increasing frame numbers on VILA-1.5-13B.

Method	FLOPs (TB)	MVBench	MLVU	VideoMME
w/FastV	78.0	50.4	52.0	48.9
w/VisionZip	56.7	48.9	50.3	47.9
w/DivPrune	37.2	**50.9**	52.5	49.4
w/METok	**36.5**	**50.9**	**52.8**	**49.9**

Table 7: Comparison of token-reduction methods with VILA-1.5-13B under 32-frame inputs.

D.2 Key Event and Frame Retention Ratios.

We conduct a joint ablation on the key event ratio α and key frame ratio β to study their impact on both efficiency and performance with LLaVA-OneVision-7B. As shown in Table 9, lower values of α and β reduce FLOPs more aggressively but lead to noticeable drops in accuracy, particularly on VideoMME. In contrast, increasing α beyond 0.5 provides marginal performance gains while incurring higher computation cost.

The best trade-off is achieved at $\alpha = 0.5$ and $\beta = 0.4$, which offers a 70% FLOPs reduction compared to the base model and consistently outperforms other settings. These results suggest that METok is stable across a reasonable range of hyper-

Method	FLOPs (TB)	AR	NQA	TR	PQA	AO	AC	ER	Overall
VILA-1.5-13B	84.1	**52.5**	53.0	78.3	57.1	35.9	**23.6**	**52.0**	50.4
w/FastV	49.8	52.0	52.4	78.7	56.4	35.0	22.0	50.0	49.5
w/VisionZip	29.1	46.5	51.3	72.2	50.7	**36.9**	22.0	50.0	47.1
w/DivPrune	20.3	51.4	54.6	78.7	57.2	35.8	22.2	51.9	50.3
w/METok (Ours)	**19.6**	50.6	**55.3**	**78.8**	**57.8**	**36.9**	22.5	51.3	**50.5**

Table 8: Results on MLVU subsets with VILA-1.5-13B.

parameters, and its performance does not rely on fine-tuned thresholds. Notably, even with $\alpha = 0.3$ and $\beta = 0.2$, METok still maintains solid accuracy, highlighting the flexibility of our key visual-text semantic identification design.

α	β	**FLOPs (TB)**	**Egoschema**	**VideoMME**
Base Model (32 frames)		71.4	43.5	58.2
0.3	0.2	14.6	42.8	57.3
0.3	0.4	17.9	43.3	57.6
0.5	0.2	17.2	43.7	58.2
0.5	0.4	21.8	44.4	58.4
0.7	0.2	34.5	44.3	58.2
0.7	0.4	43.3	44.4	58.7

Table 9: Ablation study on key event ratio α and key frame ratio β using METok on LLaVA-OneVision-7B with 32-frame input.

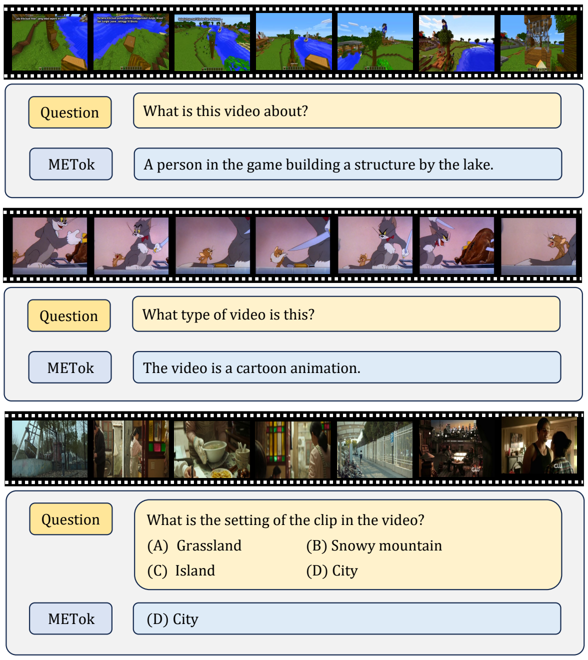
D.3 Performance across MLVU subsets

To provide a more comprehensive evaluation, we conducted a detailed analysis across all seven MLVU subcategories. As shown in Table 8, METok achieves the highest or competitive accuracy in most task types and consistently outperforms all baselines in overall accuracy and FLOPs reduction (from 84.1 to 19.6 TB). It particularly excels in long-context reasoning tasks such as NQA and PQA, where its token compression effectively preserves key visual-text semantics over time. These results not only demonstrate the task-level robustness and generality of METok, but also highlight its strength in long-form video question answering.

E Additional Qualitative Examples

We provide additional video question answering examples from video understanding benchmarks like MLVU, as shown in Figure 9 and Figure 10. In the background recognition example, METok with LongVA-7B correctly selects “Windmills” as the scene behind the engineer working with drawings, indicating robust scene-level understanding under distractor options. In the object retrieval case from an animated domain, METok accurately identifies that the cartoon lobster lifts paper money, showing resilience to style shifts and the ability to ground fine object semantics. For an attribute recognition query, METok answers white for the flower color, demonstrating precise grounding of low-level visual attributes. Together, these cases span scene context, object semantics, and fine-grained attributes, and they align with our design goal of preserving key visual-text cues while suppressing redundancy in long videos.

These examples further demonstrate the effectiveness of the proposed METok framework in capturing key visual-semantic cues from long videos. METok consistently produces precise and contextually grounded answers, successfully identifying fine-grained actions, object interactions, and temporal dependencies. By focusing on the relevant visual segments and filtering out redundant frames, METok generates accurate and coherent responses.


Question: What is this video about?

METok: A person in the game building a structure by the lake.

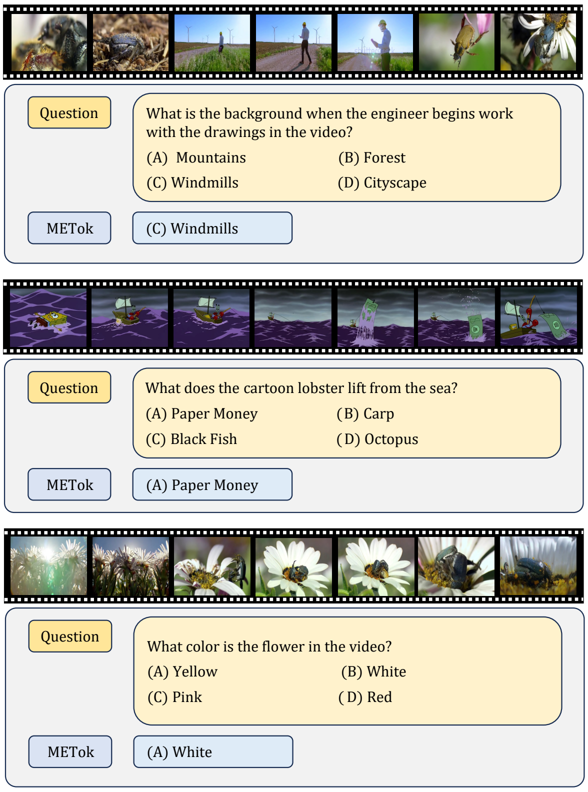
Question: What type of video is this?

METok: The video is a cartoon animation.

Question: What is the setting of the clip in the video?
(A) Grassland (B) Snowy mountain
(C) Island (D) City

METok: (D) City

Figure 9: More video understanding example of our proposed method METok with VILA-1.5-13B.



Question

What is the background when the engineer begins work with the drawings in the video?

(A) Mountains (B) Forest

(C) Windmills (D) Cityscape

METok

(C) Windmills

Question

What does the cartoon lobster lift from the sea?

(A) Paper Money (B) Carp

(C) Black Fish (D) Octopus

METok

(A) Paper Money

Question

What color is the flower in the video?

(A) Yellow (B) White

(C) Pink (D) Red

METok

(A) White

Figure 10: More video question answering example from the MLVU (Zhou et al., 2024) of our proposed method METok with LongVA-7B.