AdaTP: Attention-Debiased Token Pruning for Video Large Language Models

Fengyuan Sun[1*] Leqi Shen[1*] Hui Chen[2]
Sicheng Zhao[2] Jungong Han[3] Guiguang Ding[1,2]
[1]School of Software, Tsinghua University [2]BNRist, Tsinghua University
[3]Department of Automation, Tsinghua University

Abstract

Video Large Language Models (Video LLMs) have achieved remarkable results in video understanding tasks. However, they often suffer from heavy computational overhead due to the large number of visual tokens generated from multiple video frames. Existing visual token compression methods often rely on attention scores from language models as guidance. However, these scores exhibit inherent biases: global bias reflects a tendency to focus on the two ends of the visual token sequence, while local bias leads to an over-concentration on the same spatial positions across different frames. To address the issue of attention bias, we propose **A**ttention-**D**ebi**a**sed **T**oken **P**runing for Video Large Language Models (**AdaTP**), a novel token pruning pipeline for Video LLMs. AdaTP integrates two dedicated debiasing modules into the pipeline, targeting global attention bias and local attention bias, respectively. Without the need for additional training, our method significantly reduces the computational overhead of Video LLMs while retaining the performance of vanilla models. Extensive evaluation shows that AdaTP achieves state-of-the-art performance in various commonly used video understanding benchmarks. In particular, on LLaVA-OneVision-7B, AdaTP maintains performance without degradation while using only up to 27.3% FLOPs compared to the vanilla model. Our code will be released soon.

1 Introduction

Video large language models (Video LLMs) (Li et al., 2024; Lin et al., 2023; Wang et al., 2024b; Zhang et al., 2024c; Maaz et al., 2023) have shown potential and effectiveness in video comprehension domain. Building upon architectures of previous Multi-Modal Large Language Models (Li et al., 2023; Liu et al., 2024b; Zhu et al., 2023; Team et al., 2023; Alayrac et al., 2022; Liu et al., 2024a, 2023) designed primarily for image tasks, Video LLMs transforms raw videos into sequences of visual tokens in frame-by-frame manner. After alignment, these visual tokens are fed into the language model alongside text tokens.

Video LLMs face significant computational challenge, which can be mainly attributed to the large number of visual tokens and the quadratic complexity of attention mechanism. To mitigate this, some previous works (Chen et al., 2024a; Shang et al., 2024; Wang et al., 2024a; Yang et al., 2024; Shen et al., 2024; Tao et al., 2024; Shen et al., 2025a,b) have sought to alleviate this through visual token compression, typically leveraging attention scores to guide token pruning or merging. However, these works often fail to accurately utilize attention scores for token compression process. As a result, the selected tokens may fail to retain a comprehensive representation of the visual content, and thus lead to obvious performance degradation.

In this work, we first identify that the attention scores in the language models of Video LLMs are inherently biased, which can be characterized from two perspectives:

Global attention bias. Visual tokens with high attention scores tend to cluster at the beginning or end of the sequence, potentially overlooking intermediate important visual information.

Local attention bias. We conduct frame-wise analysis, and discovers that in shallow layers, attention scores are disproportionately concentrated on a few spatial positions, leading to over-retention of those tokens and reduced visual diversity.

To provide a more intuitive understanding of attention biases, we visualize the average attention distribution in the LLaVA-OneVision-7B model (Li et al., 2024). The visualization is computed across all samples in the VideoMME (Fu et al., 2024) dataset and shown in Fig. 1. Visualization results from other layers can be found in the Appendix A.3. In Fig. 1, we can intuitively see that the red bars,

[*] Equal contribution.

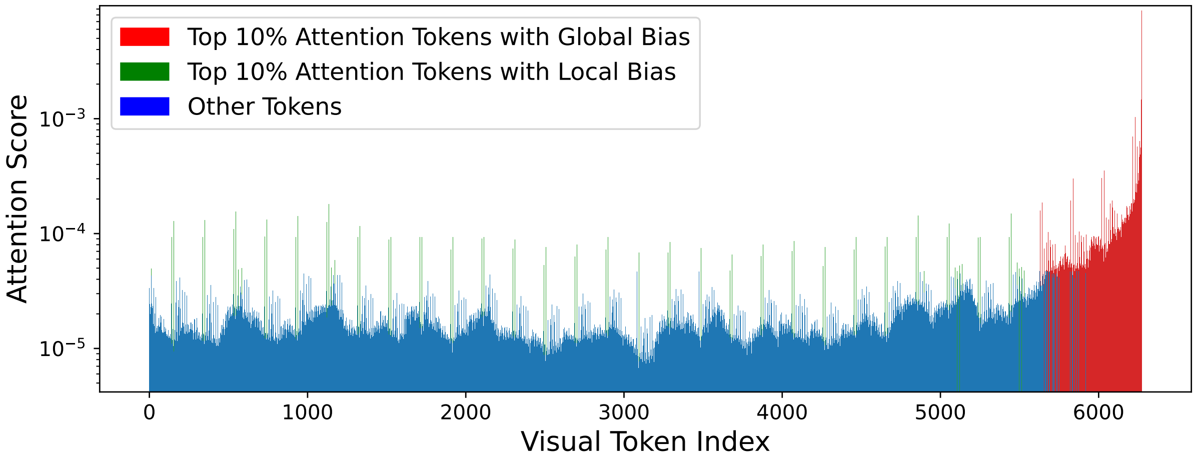

Figure 1: Visualization of attention bias. The visual tokens with top-10% attention scores are highlighted in red and green, where red tokens represent global attention bias, and green tokens represent local attention bias.

which represent high attention scores, are concentrated in the latter part of the sequence, showing global attention bias; the green bars, exhibit a periodic pattern throughout the entire sequence, representing local attention bias.

To effectively leverage attention scores to guide visual token compression, we argue that it is essential to address the aforementioned biases. In this paper, we propose targeted modules to mitigate global and local attention bias respectively. Building upon these insights, we introduce Attention-Debiased Token Pruning for Video Large Language Models (**AdaTP**), a training-free and plug-and-play acceleration pipeline for Video LLMs. To be specific, we first partition adjacent similar frames into coherent segments, then introduce two debiasing modules: the Global Debiasing Module helps the model focus on semantically meaningful segments by identifying visual contents that are closely related to textual information; the Local Debiasing Module performs token pruning adaptively and independently within each segment, which encourages the preservation of diverse visual content and effectively reduces intra-segment redundancy.

We integrate our AdaTP pipeline into two types of representative Video LLM architectures: LLaVA-OneVision (Li et al., 2024) and LLaVA-Video (Lin et al., 2023). Experimental results show that our method achieves state-of-the-art results on several video understanding benchmarks including VideoMME (Fu et al., 2024), LongVideoBench (Wu et al., 2024), and MLVU (Zhou et al., 2024), which are widely adopted to assess the comprehension ability of Video LLMs, and contain diverse video types. On LLaVA-OneVision-7B, our method achieves comparable performance to the original model while keeping only 27.3% of its FLOPs.

Our main contributions are as follows:

1. We reveal and thoroughly analyze the attention bias phenomenon in Video LLMs from both global and local perspectives.
2. We propose AdaTP, a novel token pruning framework for Video LLMs. AdaTP mitigates attention biases and leverages debiased attention scores as an effective guidance for visual token compression, thus reduced computational burden without sacrificing models' performance.
3. Experimental results demonstrate that AdaTP achieves state-of-the-art performance on diverse video comprehension benchmarks without requiring additional training.

2 Related Works

2.1 Video Large Language Models

With the success of Multi-modal Large Language Models (MLLMs), increasing efforts have been made to extend their capabilities to video understanding. Video-LLaVA (Lin et al., 2023) aligns images and videos, allowing language models to learn a unified visual representation, enabling language models to understand images and videos content simultaneously. Qwen-2-VL (Wang et al., 2024b) extends dynamic resolution and updates mRoPE in the time dimension, enhancing the model's ability to learn from temporal sequence. PLLaVA (Xu et al., 2024) applies pooling strategy to smooth the feature distribution along the temporal dimension and extended Image LLMs. LLaVA-OneVision (Li et al., 2024) unifies image and video under a single framework for consistency and effectiveness. LLaVA-Video (Zhang et al., 2024c) extends visual instruction tuning to videos, training model with synthetic dataset for video instruction-following, and introduces newline tokens for each frame to distinguish spatial and temporal positions. InternVL-2.5 (Chen et al., 2024b) uses pixel unshuffle operation to reduce the number of visual tokens.

LLaVA-OneVision (Li et al., 2024) and LLaVA-Video (Zhang et al., 2024c) are two of the most widely used and representative Video LLMs, thus we choose to evaluate our method on these two model architectures.

2.2 Visual token compression

Token compression methods have been proposed to accelerate visual encoder and MLLM's inference speed. In earlier years, several works (Rao et al., 2021; Yin et al., 2022; Liang et al., 2022; Marin et al., 2021) were done to explore token compression for Vision Transformers (ViT). Recent studies propose visual token compression methods to accelerate inference in MLLMs and Video LLMs. FastV (Chen et al., 2024a) reveals the high redundancy of visual tokens in language models and leverages attention scores from it to guide the pruning process. LLaVA-PruMerge (Shang et al., 2024) and several subsequent works (Wang et al., 2024a; Zhang et al., 2024b) perform token pruning and merging based on the attention scores of [CLS] token from the visual encoder. VisionZip (Yang et al., 2024) selects a small number of key visual tokens based on the attention information from the visual encoder, and applies a merging strategy to retain the remaining information. Dycoke (Tao et al., 2024) is specifically tailored for Video LLMs, which performs fixed-length segmentation of videos before feeding them into the LLM, removes and merges redundant parts, and further prunes tokens during the decoding stage of the LLM.

Most of the aforementioned methods rely on attention scores to guide visual token compression and have achieved promising results. However, their overall effectiveness remains limited. These methods often suffer from notable performance degradation due to attention bias. In contrast, our method mitigates attention bias through a targeted debiasing mechanism, enabling more efficient and effective visual token compression.

3 Methodology

3.1 Preliminaries

Video LLM's Inference. Video LLM's inference process can be divided into the following steps: a video is processed by visual encoder (such as CLIP (Radford et al., 2021) and SigLIP (Zhai et al., 2023)) in frame-by-frame paradigm. After alignment through a projector, the visual tokens are concatenated with tokenized text tokens and system prompt tokens, serve together as the input to the LLM. We denote $VE(\cdot)$ as the visual encoder that processes the video v, and $P(\cdot)$ as the projector alignment process. The processed visual tokens $P(VE(v))$ can be represented as:

$$X_v^0 = P(VE(v)) \in \mathbb{R}^{nc \times d}, \tag{1}$$

where X_v^0 is the output of the visual projector and serves as the input to the 0-th layer of the LLM, while X_v^l denotes the input to the l-th layer. X_v^l can also be represented as $[X_{v_{f_1}}^l, \ldots, X_{v_{f_n}}^l]$ in frame-by-frame manner, and $X_{v_{f_i}}^l (i = 1, 2, \ldots, n)$ is denoted as visual tokens of the i-th frame from the video in the l-th layer, where n is the frame count. c is the number of tokens within any single frame, and d is the hidden dimension of visual tokens.

Attention score in LLM as token importance metric. In Video LLMs, attention scores produced by LLMs can serve as a metric for token importance. In the self-attention module of the l-th layer in language models, by averaging across all attention heads, we get attention map $A^l \in \mathbb{R}^{L \times L}$. We extract the attention scores from text tokens to visual tokens by averaging the rows of the attention matrix A^l corresponding to text tokens and slicing the columns corresponding to visual tokens, resulting in $s^l \in \mathbb{R}^{nc}$. In the $(l+1)$-th layer, these scores can be used to guide the token compression process. However, directly selecting the visual tokens with the highest attention scores leads to significant performance degradation, as will be discussed in the next subsection.

3.2 Observation: Attention bias phenomenon

Many existing works (Zhang et al., 2023; Xiao et al., 2023; Chen et al., 2024a; Tao et al., 2024) leverage attention scores to guide token pruning. However, Such strategies are actually suboptimal

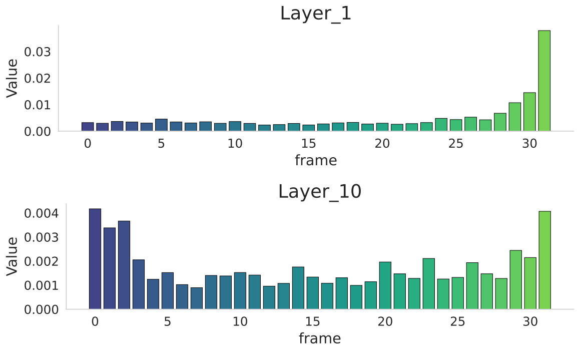

Figure 2: Visualization of global attention bias in Layer 1 and 10. Each bar represents the sum of attention scores of visual tokens within a single frame.

due to the presence of attention bias, leaving substantial room for improvement. The two parts of attention biases are stated as follows:

Observation 1. **Global attention bias.** Ideally, to serve as a visual token importance metric, for specific text-video pairs, attention scores should be concentrated on tokens from frames that are highly relevant to the text. However, in practice, in almost all layers, these tokens are predominantly located in the beginning and the final portion of the sequence, regardless of the content of text-video pair. In Fig. 2, each bar represents the sum of attention scores of all visual tokens within a single video frame. This figure clearly illustrates the presence of global bias, as the attention scores are concentrated at the first and last few frames, while the middle frames receive significantly less attention. We take attention distribution of layer 1 as an example, among the tokens with the top 10% attention scores, 86.8% are concentrated in the last 4 frames out of total 32 frames, which clearly shows the global attention bias.

Observation 2. **Local attention bias.** In shallow layers of language models, we observe that across all frames of videos, Video LLMs intrinsically allocate substantial attention to a few specific spatial positions, and remarkably, these spatial positions remain fixed regardless of variations in video and text content. In Fig. 3, each grid shows the total attention received by that spatial position across all frames, and the 196 grids correspond to the 14×14 spatial patches extracted from each video frame. In layer 1, the spatial position receiving highest attention (Row 11, Column 1) receives 5.77 times the average attention of all visual tokens, clearly illustrating the presence of local bias.

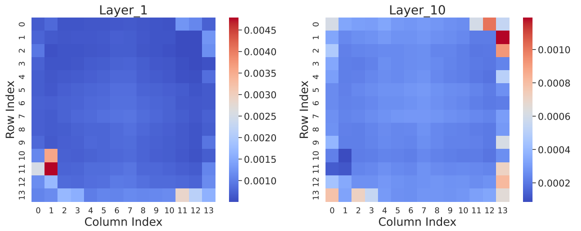

Figure 3: Visualization of local attention bias in Layer 1 and 10. Each block illustrates the sum of attention scores from prompt tokens to each spatial position across all frames.

Due to the biases stated above, directly leveraging raw attention scores as importance metric is not accurate and thus suboptimal. To achieve more precise pruning of visual tokens, we propose the method described in the next subsection. Based on the correction of the two types of biases mentioned above, we can effectively leverage attention scores to guide the token pruning process.

3.3 Our pipeline: AdaTP

Based on the above observations, we propose a new token pruning pipeline named Attention-Debiased Token Pruning for Video LLMs (AdaTP).

Due to temporal coherence of videos, a video can typically be divided into several parts, where frames within each part are highly similar, while different parts convey diverse content, thus we divide each video into multiple segments. For each video, its visual feature sequence $VE(v) = (v_1, v_2, \ldots, v_n)$ are partitioned into consecutive segments $\mathcal{S} = (s_1, s_2, \ldots, s_m)$ based on cosine similarity between adjacent frames, where m denotes the number of segments. Given a threshold τ_s, two adjacent frames v_i and v_{i+1} are assigned to the same segment if $cos\_sim(v_i, v_{i+1}) \geq \tau_s$. Otherwise, a new segment is started at frame v_{i+1}.

Correspondingly, X_v^l can be represented in segment-by-segment manner:

$$X_v^l = [X_{s_1}^l, X_{s_2}^l, \ldots, X_{s_m}^l]. \tag{2}$$

Inspired by AIM (Zhong et al., 2024) and PyramidDrop (Xing et al., 2024), we adopt progressive token pruning paradigm. In shallow layers of the language model, we gradually prune a certain number of visual tokens at each layer.

As show in Fig. 4, our pipeline mainly consists of two modules: Global Debiasing Module and Local Debiasing Module. The Global Debiasing Module leverages the text encoder to identify signif-

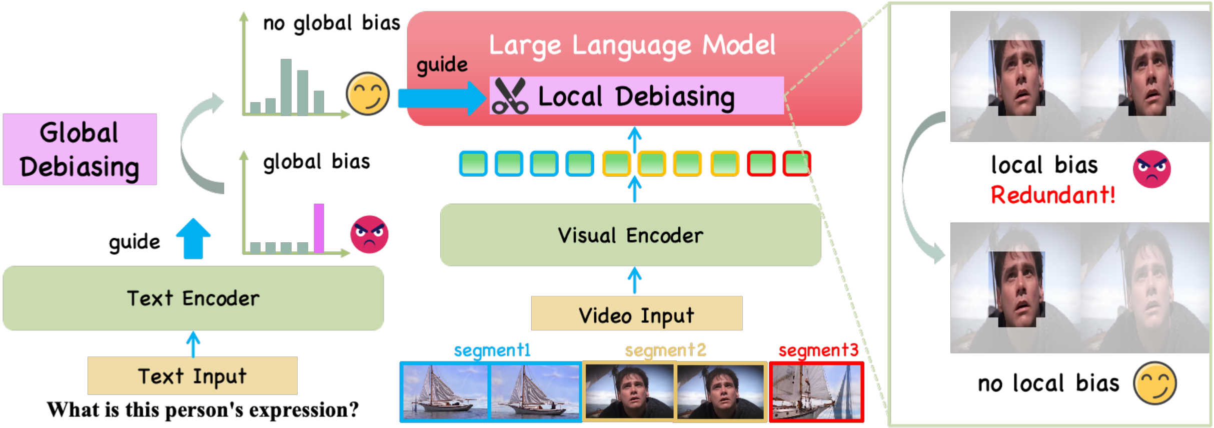

Figure 4: Illustration of our AdaTP pipeline. AdaTP mitigates attention bias by selectively retaining critical tokens based on text relevance and spatial diversity. The Global Debiasing Module identifies significant video segments, and the Local Debiasing Module further alleviates intra-segment spatial redundancy.

icant segments which are highly relevant to textual information, and assigns higher retention ratios to them. The local debiasing module operates within each segment, removing redundant visual tokens that occupy the same spatial positions across these adjacent and similar frames.

(1) **Global Debiasing Module.** We introduce the text encoder aligned with the visual encoder in Video LLMs to tackle global attention bias. For i-th video frame, the visual encoder produces a global visual token $v_{pi} \in \mathbb{R}^d$. Meanwhile, the input text is processed by the text encoder $TE(\cdot)$ to obtain a global textual token $t_p \in \mathbb{R}^d$. Text relevance of each frame can then be measured by computing the cosine similarity between v_{pi} and t_p:

$$sim_i = \frac{v_{pi}^\top t_p}{\|v_{pi}\| \cdot \|t_p\|}, i = 1, 2, \ldots, n, \quad (3)$$

where $\|\cdot\|$ denotes the L2 norm. We estimate the text relevancy of each segment by averaging similarities of all frames within it. We then introduce a threshold parameter τ_t, and segments with similarity exceeding τ_t are classified as highly relevant to the text (hereafter referred to as significant segments for convenience). The selected significant segments can be represented as set $\mathcal{S}' \subseteq \mathcal{S}$.

We perform token pruning within each segment, assigning higher retention ratios to significant ones. To be specific, we define an overall compression ratio r, retaining totally $L_v^l \cdot r$ visual tokens, distributed across segments. For each segment s, the number of retained tokens is proportional to its frame count $\|s\|$. For significant segments $s_i \in \mathcal{S}'$, this allocation is upscaled by a factor α_{boost}, but constrained such that the total retained tokens in all significant segments do not exceed $\gamma_{\text{cap}} \cdot L_v^l$, ensuring coherence and preventing over-concentration. This strategy effectively preserves critical content while reducing redundancy.

The retention ratio of significant segments are:

$$r_1 = \min\left(\alpha_{\text{boost}}, \frac{n}{\sum_{s \in \mathcal{S}'} \|s\|} \cdot \gamma_{\text{cap}}\right) \cdot r, \quad (4)$$

while the retention ratio of other segments are:

$$r_2 = \frac{r \cdot n - r_1 \cdot \sum_{s \in \mathcal{S}'} \|s\|}{\sum_{s \notin \mathcal{S}'} \|s\|}. \quad (5)$$

Within each segment s_i, we select the above ratio of visual tokens (r_1 for significant segments, and r_2 for others) with the highest attention scores.

Algorithm 1: Local Debiasing Module

```
Input: $V = \{v_1, v_2, \ldots, v_n\}$: selected
       visual tokens within a segment.
pos($v_i$): spatial position of token $v_i$;
score($v_i$): attention score of token $v_i$.
Output: $V' \subseteq V$: deduplicated token set.
Sort $V$ in descending order by score($v_i$);
$P \leftarrow \emptyset$;   // Used spatial positions
$V' \leftarrow \emptyset$;   // Selected tokens
for $v_i$ in $V$ do
    if $pos(v_i) \notin P$ then
        Add $v_i$ to $V'$;
        Add pos($v_i$) to $P$;
return $V'$
```

Table 1: Comparison of our method with other baselines across various benchmarks on three Video LLMs. Within the same comparison, the highest compression ratio and highest performances are highlighted in bold.

Model	Method	FLOPs	VideoMME				MLVU	LongVideoBench	Avg
			short	medium	long	overall			
LLaVA-OneVision-0.5B	vanilla	100.00%	53.4	40.9	36.9	43.74	46.45	47.64	45.94
	FastV$^{\text{(ECCV'24)}}$	55.19%	53.1	**42.1**	36.6	43.93	45.07	45.77	44.92
	VisionZip$^{\text{(CVPR'25)}}$	44.38%	53.4	40.7	37.4	43.85	47.23	47.12	46.07
	Dycoke$^{\text{(CVPR'25)}}$	50.32%	55.0	40.4	**38.7**	**44.70**	47.24	46.97	46.30
	AdaTP (ours)	**41.08%**	**55.4**	41.8	36.4	44.56	**47.36**	**47.79**	**46.57**
	FastV	43.66%	52.3	**41.2**	36.4	43.33	45.38	45.85	44.85
	VisionZip	39.37%	53.8	41.0	37.0	43.93	**47.54**	46.15	45.87
	Dycoke	36.50%	54.4	40.3	**37.7**	44.10	45.26	46.37	45.24
	AdaTP (ours)	**33.99%**	**55.8**	**41.2**	36.4	**44.48**	47.25	**48.24**	**46.66**
	FastV	32.13%	48.0	41.4	36.7	42.04	44.47	45.03	43.85
	VisionZip	32.36%	52.6	40.2	36.1	42.96	46.12	46.30	45.13
	Dycoke	32.86%	53.9	39.2	36.1	43.07	42.23	44.50	43.27
	AdaTP (ours)	**26.43%**	**54.7**	**42.7**	**37.6**	**44.96**	**47.58**	**47.05**	**46.53**
LLaVA-OneVision-7B	vanilla	100.00%	70.0	56.7	48.7	58.44	63.25	56.32	59.34
	FastV	54.64%	70.0	55.0	47.2	57.41	62.22	55.65	58.43
	VisionZip	46.38%	**70.9**	**56.9**	49.0	**58.93**	63.22	55.65	59.27
	Dycoke	50.32%	70.1	54.2	48.7	57.67	63.16	56.39	59.39
	AdaTP (ours)	**43.00%**	70.4	56.2	**49.4**	58.70	**63.95**	**56.69**	**59.78**
	FastV	43.18%	68.6	55.1	47.9	57.19	61.35	55.12	57.89
	VisionZip	39.37%	**71.1**	55.8	**49.7**	58.85	63.09	56.02	59.32
	Dycoke	36.44%	67.6	53.3	48.0	56.30	61.13	**56.69**	58.04
	AdaTP (ours)	**35.48%**	71.0	**56.8**	48.9	**58.89**	**64.27**	55.95	**59.70**
	FastV	35.20%	66.1	53.6	48.1	55.93	60.82	51.91	56.22
	VisionZip	27.95%	69.2	**57.7**	**49.2**	58.70	63.13	**56.17**	59.33
	Dycoke	32.84%	67.1	53.1	46.6	55.59	60.90	55.95	57.48
	AdaTP (ours)	**27.30%**	**71.0**	56.9	**49.2**	**59.04**	**63.47**	56.02	**59.51**
LLaVA-Video-7B	vanilla	100.00%	70.0	56.7	48.7	58.44	63.25	56.32	59.34
	FastV	54.89%	69.4	59.2	**50.2**	59.63	61.56	54.15	58.45
	VisionZip	44.42%	72.6	58.9	48.3	59.93	61.55	55.72	59.07
	Dycoke	59.26%	72.2	57.7	47.8	59.22	60.67	55.57	58.49
	AdaTP (ours)	**44.26%**	**72.7**	**59.9**	49.9	**60.81**	**62.10**	**56.32**	**59.74**
	FastV	43.61%	67.2	57.1	47.9	57.41	60.00	52.81	56.74
	VisionZip	37.52%	71.0	58.1	49.3	59.48	61.34	54.97	58.60
	Dycoke	38.71%	68.9	56.7	47.8	57.78	58.83	54.23	56.95
	AdaTP (ours)	**36.63%**	**72.2**	**59.8**	**50.0**	**60.67**	**61.42**	**56.77**	**59.62**
	FastV	32.33%	62.6	54.1	48.0	54.89	57.80	51.09	54.59
	VisionZip	28.69%	**70.4**	57.8	49.3	59.19	59.81	54.08	57.69
	Dycoke	35.36%	68.1	55.7	46.2	56.67	59.02	53.78	56.49
	AdaTP (ours)	**28.27%**	70.3	**59.1**	**51.7**	**60.37**	**60.95**	**55.72**	**59.01**

(2) **Local Debiasing Module.** Due to the similarity among frames within the same segment, visual tokens from identical spatial positions among the selected high-attention tokens are redundant. Our Local Debiasing Module removes spatially redundant visual tokens, thereby further reducing unnecessary computational overhead.

We propose a simple yet effective strategy to address this redundancy. Specifically, after obtaining the set of visual tokens with the highest attention scores within each segment, we sort them in descending order based on their scores. We then iterate through the sorted list, and for each token, we check whether another token from the same spatial position in other frames has already been selected. If so, the current token is considered redundant and discarded; otherwise, it is retained. This ensures that only one representative token is kept for each spatial location within the segment. The full procedure is detailed in Algorithm 1.

Moreover, with this design, our pipeline fully accounts for diverse and unique characteristics of each video by adaptively adjusting the visual token compression rate. Rather than applying a fixed pruning scheme, more dynamic videos—those exhibiting frequent scene changes or significant motion—are typically divided into more segments, allowing them to retain a larger proportion of visual tokens. In contrast, static videos with minimal variation across frames are partitioned into fewer

Table 2: Ablation study on different modules. Seg. means segment-aware token pruning, Global.D denotes Global Debiasing Module, while Local.D denotes Local Debiasing Module. ✓ indicates the bias is corrected; and ✗ indicates it is not.

Seg.	Global.D	Local.D	FLOPs	VideoMME	MVLU	LongVideoBench	Avg
✗	✗	✗	27.65%	43.67	46.58	44.88	45.04
✓	✗	✗	27.76%	44.63	46.26	46.67	45.85
✓	✓	✗	27.33%	44.63	47.24	**47.19**	46.35
✓	✓	✓	**26.43%**	**44.96**	**47.58**	47.05	**46.53**

segments and accordingly allocated fewer tokens. This ensures that token pruning is more aligned with the content dynamics of the videos.

4 Experiments

4.1 Implementation Details

Benchmarks. We evaluate the performance of Video LLMs on three widely used video understanding benchmarks: VideoMME (Fu et al., 2024), MLVU (Zhou et al., 2024), and LongVideoBench (Wu et al., 2024). The VideoMME dataset is divided into short (1-3 minutes), medium (3-30 minutes), and long subsets (30-60 minutes), each containing 900 video-question pairs. The MLVU dataset consists of videos ranging from 3 minutes to 2 hours in length and includes nine distinct evaluation tasks, totaling 2,174 samples. LongVideoBench comprises video-question pairs across 17 categories, ranging from 1 to 60 minutes; we use its validation set, which contains 1,337 samples. These datasets cover various types, durations, and scenarios of videos, providing a fair and comprehensive evaluation for video understanding ability of Video LLMs.

Model Settings. We apply our method on three types out of two representative Video LLM architectures: LLaVA-OneVision-0.5B, LLaVA-OneVision-7B, (Li et al., 2024) and LLaVA-Video-7B (Lin et al., 2023). Under our settings, LLaVA-OneVision models samples 32 frames, generates 196 visual tokens per frame; LLaVA-Video model samples 20 frames, generates 182 visual tokens per frame, including 13 added newline tokens. All experiments are conducted on a single NVIDIA 3090 GPU. All FLOPs statistics are collected using the torch.profiler package provided by PyTorch for consistent and accurate measurement.

Our progressive pruning process starts from the second layer of the language model until the $(N-12)$-th layer, where N denotes the total number of layers in language model ($N = 24$ for LLaVA-OneVision-0.5B model, and $N = 28$ for LLaVA-OneVision-7B and LLaVA-Video-7B models). Before passing to the second layer, we retain $p \cdot 40\%$ of the visual tokens; after layer $N-12$, only $p \cdot 12\%$ of the tokens are kept; for all intermediate layers, tokens are removed at a uniform rate. Here p is the parameter for us to adjust the compression ratio. For hyperparameters of our pipeline modules, we set $\tau_t = 0.05, \alpha_{\text{boost}} = 2.2, \gamma_{\text{cap}} = 0.75, \tau_s = 0.95$ for all models. For three different compression ratios, we set p to 2.0, 1.5 and 1.0, respectively.

Baselines. We adopt three training-free visual token compression methods: FastV, VisionZip and Dycoke. We use official codes of these methods for fair comparison, and conduct all evaluations using LMMs-Eval (Zhang et al., 2024a). Detailed settings please refer to Appendix A.1.

4.2 Main Results

Tab.1 compares our AdaTP with state-of-the-art methods on aforementioned models. On all three models and with three compression ratios, our AdaTP outperforms state-of-the-art baselines. On both LLaVA-OneVision-0.5B and LLaVA-OneVision-7B, our method surprisingly surpasses the vanilla model. We attribute this improvement to our pipeline's ability to eliminate redundant visual information and guide the model to focus more on critical tokens. On LLaVA-Video-7B, our AdaTP also showed clear advantage over other baselines.

As the compression ratio increases, all other methods experience a significant performance drop, whereas our method consistently maintains strong performance. This shows the robustness and effectiveness of our AdaTP.

4.3 Ablation Studies

We conduct ablation studies on modules in our pipeline to prove their functionality, and also on

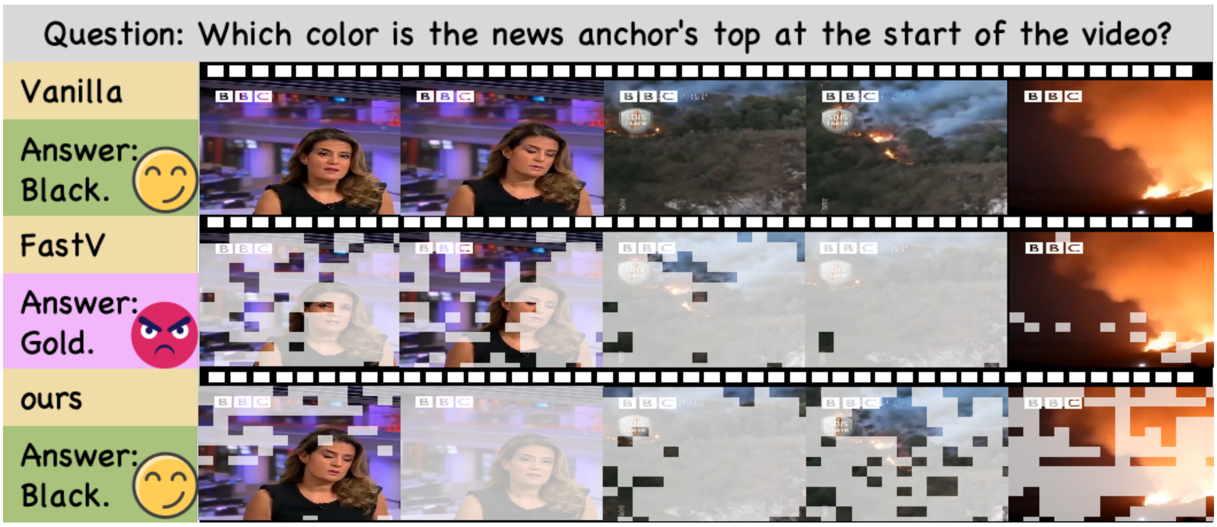
Question: Which color is the news anchor's top at the start of the video?
Vanilla
Answer: Black.
FastV
Answer: Gold.
ours
Answer: Black.

Figure 5: Visualization example sampled from the VideoMME dataset. We adopt attention scores from layer 1 to generate the visualization results, in which we perform significant token pruning.

hyperparameters to show its robustness. To ensure consistency and clarity in our presentation, we conduct the ablation study on LLaVA-onevision-0.5B, and p is set to 1.

Ablation study on debiasing modules. We first conduct ablation studies on our two debiasing modules and our segment-aware design. Without Local Debiasing module, intra-segment spatial redundancy will be retained; without the Global Debiasing Module, significant segments will not be distinguished. The results demonstrate the effectiveness of segment-aware design and both of our two debiasing modules, further highlight the impact of attention bias on the performance of pruned Video LLMs. The results are shown in Tab. 2.

Ablation study on hyperparameters. We evaluate the performance of hyperparameters $\tau_s, \tau_t, \alpha_{\text{boost}}$, and γ_{cap}. In this part, we also set p to 1. The results are shown in Appendix A.2.

For progressive layer-by-layer token pruning, we start pruning from Layer 2 while keeping the last 12 layers unpruned across all Video LLMs. A uniform pruning rate is applied between the pruned layers, with a fixed number of tokens discarded after each layer. In LLaVA-OneVision-0.5B (Li et al., 2024), which consists of 24 layers, pruning depth is by default set to Layer 2-12. Variants with different pruning depths are also presented in Tab. 3 for comparison.

4.4 Visualization

To further illustrate the effectiveness of our method, we visualize the token pruning process. The pruned tokens are visually masked with a white overlay to indicate their exclusion. As shown in Fig. 5, for this video-question pair, the vanilla model produces the correct answer, while FastV gives an incorrect response. In contrast, our method retains significant visual tokens and answers correctly. It can be observed that due to the influence of global bias, FastV fails to extract text-relevant content effectively and retains a large number of visual tokens from the latter part of the sequence. Our approach, however, accurately identifies the video segments closely aligned with the textual semantics, *i.e.* the first two frames containing the news anchor. Among these two visually similar frames, our method retains informative content from only one frame, thereby effectively addressing the visual redundancy caused by local bias.

Table 3: Ablation study on pruning layer depths.

layers	FLOPs	VideoMME	MLVU	LongVideoBench	Avg
2-8	24.46%	44.30	47.12	46.30	45.91
2-10	25.45%	44.74	47.90	46.60	46.41
2-12	26.43%	44.96	47.58	47.05	46.53
2-14	27.37%	44.52	46.89	47.57	46.33
2-16	28.40%	44.78	47.33	47.57	46.56

5 Conclusion

In this paper, we identified and thoroughly analyzed the attention bias phenomenon in Video LLMs, and proposed AdaTP, a training-free token pruning pipeline. Our AdaTP effectively addressed the issue of attention bias, relieved the heavy computational burden of Video LLMs while retaining critical visual contents. Extensive experiments on different Video LLMs and video understanding benchmarks have demonstrated the effectiveness and robustness of our method.

Limitations

Due to computational resource constraints, our proposed method has not yet been validated on larger-scale Video LLMs. In addition, our pipeline introduces a relatively large number of hyperparameters, which may increase its complexity. However, we have conducted comprehensive ablation studies on them, demonstrating that the method remains robust across these hyperparameters.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

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.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

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

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*.

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*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llavanext: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. 2021. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949.

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. 2025a. Fastvid: Dynamic density pruning for fast video large language models. *arXiv 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*.

Leqi Shen, Tao He, Guoqiang Gong, Fan Yang, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. 2025b. Llava-mlb: Mitigating and leveraging attention bias for training-free video llms. *arXiv preprint arXiv:2503.11205*.

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*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Ao Wang, Fengyuan Sun, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. 2024a. [cls] token tells everything needed for training-free efficient mllms. *arXiv preprint arXiv:2412.05819*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. 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. *Advances in Neural Information Processing Systems*, 37:28828–28857.

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 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. 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*.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2022. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818.

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 1 others. 2024a. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2024b. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*.

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

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

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*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Appendix

A.1 Baseline Settings

For VisionZip, we set the ratio of dominant visual tokens and contextual visual tokens to 27:5 following the default setting. In LLaVA-Video (Zhang et al., 2024c), to solve the existence of newline tokens, after visual encoder, we first applied original method in VisionZip, then added newline tokens following, making sure the ratio of compressed visual tokens and added newline tokens remains fixed.

For Dycoke, L is set to 3, P is set to 0.7 following the default settings, and K is set to 0.7, 0.9 and 0.95 for fair comparison with three compression ratios $p = 2.0, 1.5, 1.0$ adopted in our experiments.

For FastV, following the default settings, we couduct token pruning at layer 2 for all Video LLMs. The visual token retention ratio is set to $0.50, 0.375$ and 0.25 for three compression ratios $p = 2.0, 1.5, 1.0$ adopted in our experiments.

A.2 Ablation studies on hyperparameters

τ_t sets the threshold for significant segments. Values that are too high or too low can lead to segments more relevant to the text not being accurately distinguished; α_{boost} and γ_{cap} control the number of tokens to keep within each significant segment. We properly adjusted their values to retain sufficient but not redundant tokens in significant segments, while keeping necessary tokens in others; τ_s influences the segment partitioning process. If too high, similar frames may be separated into different segments; if too low, segments may contain cluttered visual information and discard useful visual tokens. The results are shown in Tab. 4a–4d.

It is worth noting that our pipeline demonstrates strong performance across a wide range of hyperparameter settings, highlighting its robustness.

A.3 Additional Visualizations of attention bias phenomenon

In Fig. 1–3, we visualized the attention bias phenomenon in layer 1 and 10 from LLaVA-OneVision-7B (Li et al., 2024). More visualizations from other layers are shown in Fig. 6 and Fig. 7. It can be observed that the global and local attention bias we mentioned in this paper appear across all the layers shown, demonstrating the universality and consistency of our observation.

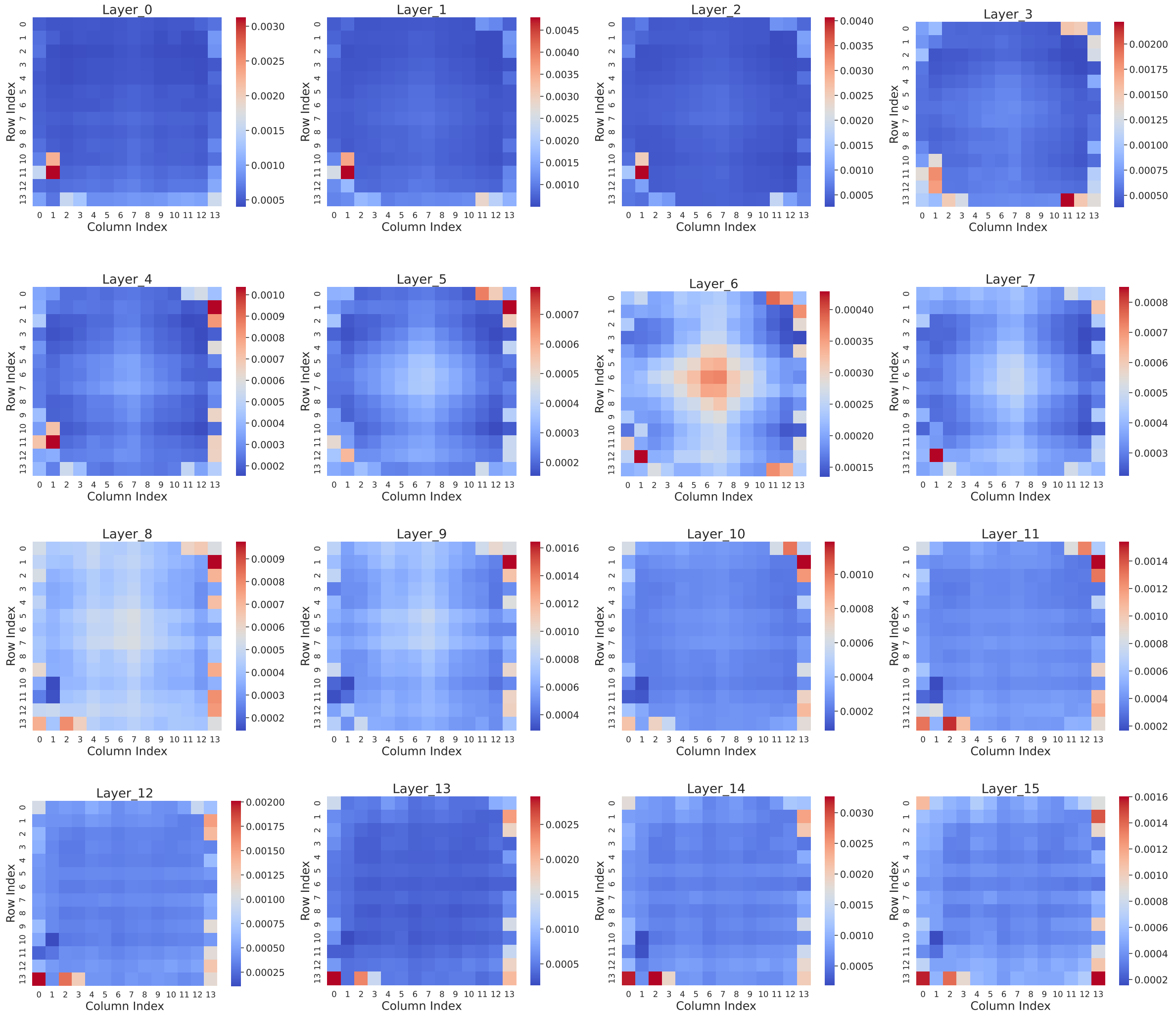

Figure 6: Additional visualizations of local attention bias across different layers.

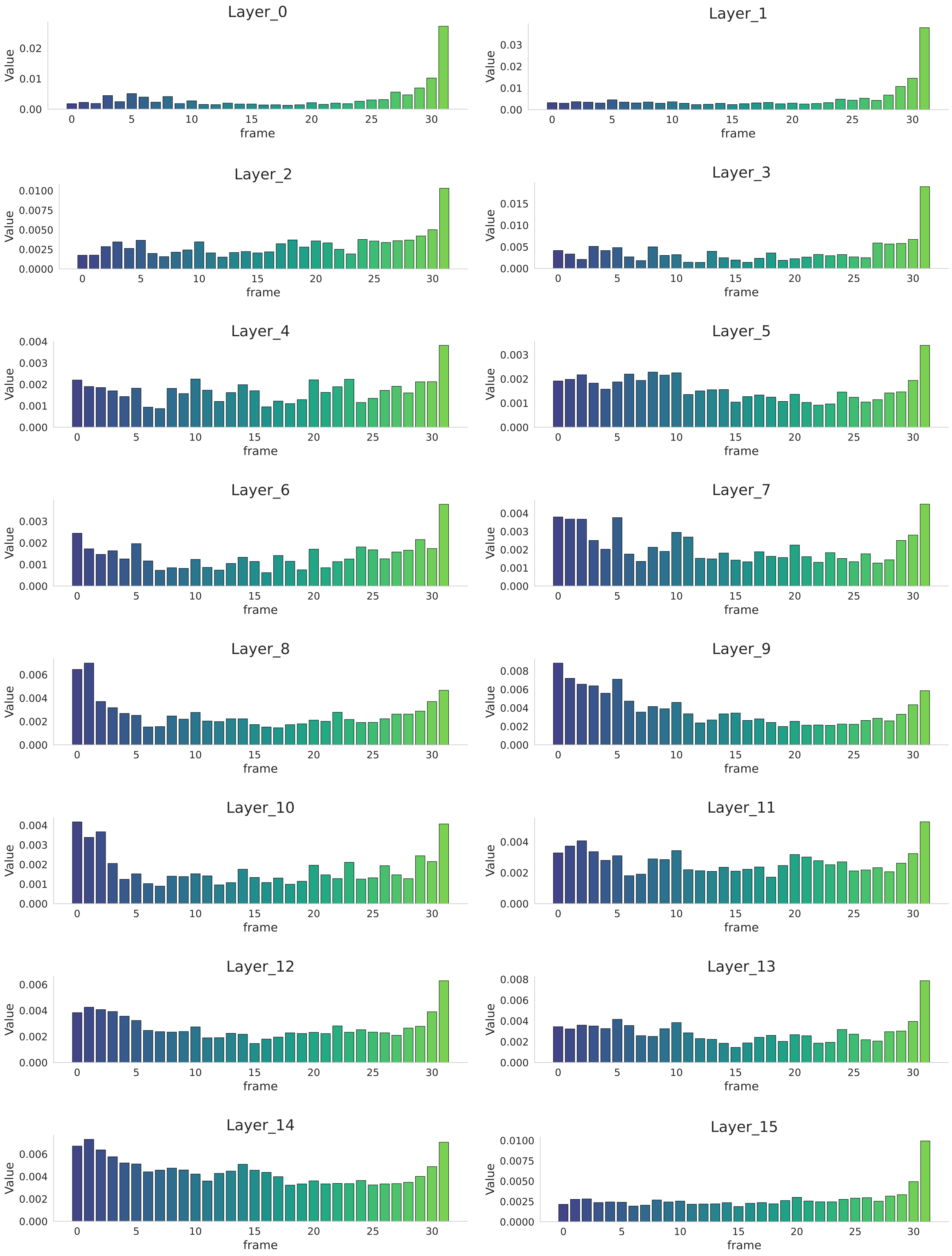

Figure 7: Additional visualizations of global attention bias across different layers.

Table 4: Ablation study on different hyperparameters.

(a) Ablation on τ_t

τ_t	VideoMME	MLVU	LongVideoBench	Avg
0.03	44.63	46.40	46.30	45.78
0.04	44.81	47.35	46.37	46.18
0.05	**44.96**	**47.58**	**47.05**	**46.53**
0.06	44.85	47.50	46.37	46.24
0.07	44.70	47.47	46.97	46.38

(b) Ablation on α_{boost}

$\alpha_{\textbf{boost}}$	VideoMME	MLVU	LongVideoBench	Avg
1.6	44.44	47.28	47.19	46.30
1.8	44.44	47.51	47.05	46.33
2.0	44.74	**47.66**	46.75	46.38
2.2	**44.96**	47.58	47.05	**46.53**
2.4	44.56	47.49	**47.27**	46.44
2.6	44.37	47.52	46.97	46.29

(c) Ablation on γ_{cap}

$\gamma_{\textbf{cap}}$	VideoMME	MLVU	LongVideoBench	Avg
0.50	44.81	46.80	**47.05**	46.22
0.75	**44.96**	**47.58**	**47.05**	**46.53**
1.00	44.89	47.23	46.82	46.35

(d) Ablation on τ_s

τ_s	VideoMME	MLVU	LongVideoBench	Avg
0.60	38.00	43.32	43.16	41.49
0.80	41.93	44.81	45.85	44.20
0.90	44.11	47.44	46.67	46.07
0.95	**44.96**	**47.58**	47.05	**46.53**
1.00	44.63	47.24	**47.19**	46.31