VisionTrim: Unified Vision Token Compression for Training-Free MLLM Acceleration

Anonymous Authors[1]

Abstract

Multimodal large language models (MLLMs) suffer from high computational costs due to excessive visual tokens, particularly in high-resolution and video-based MLLMs. Existing token reduction methods focus on isolated pipeline components and neglect textual alignment, leading to performance degradation. In this paper, we propose VisionTrim, a unified framework for training-free MLLM acceleration, integrating two plug-and-play modules: 1) Dominant Vision Token Selection (DVTS) module preserves essential visual tokens via global-local view, and 2) Text-Guided Vision Complement (TGVC) module enables context-aware token merging guided by textual cues. Experiments across diverse image and video benchmarks demonstrate the performance superiority of our VisionTrim, advancing practical MLLM deployment in real-world scenarios. Our full implementation will be publicly available.

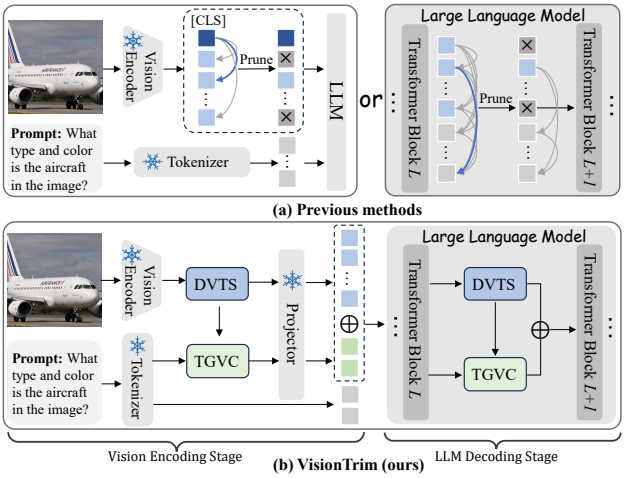


Figure 1. **Comparison between previous token pruning methods and our proposed VisionTrim.** (a) Previous methods focus solely on a specific part of MLLM framework, typically either the vision encoding or LLM decoding stage. (b) VisionTrim, in contrast, considers the entire MLLM pipeline. Specifically, two plug-and-play modules, Dominant Vision Token Selection (DVTS) and Text-Guided Vision Complement (TGVC) are introduced for token reduction in both vision encoding and LLM decoding stages.

1. Introduction

With the recent advancements in large language models (LLMs) (Vicuna, 2023; Touvron et al., 2023; Bai et al., 2023a; Achiam et al., 2023), significant efforts (Bai et al., 2023b; Chen et al., 2024b; Reid et al., 2024) have been devoted to extending their impressive reasoning and interaction capabilities to vision-language tasks. Current multimodal large language models (MLLMs) typically integrate visual signals as sequential visual tokens, which are processed by an LLM to enable visual perception of the world.

Despite their promising performance, the extensive use of visual tokens, which dominate the input sequence of LLM, substantially increases the computational complexity and cost associated with inference in MLLMs. This issue is particularly pronounced in high-resolution methods (Liu et al., 2024b;c; Chen et al., 2024b) and video-based models (Zhang et al., 2024c; Cheng et al., 2024), where the increased token length exacerbates computational overhead and severely restricts the practical deployment potential of VLMs (Jin et al., 2024).

Recent studies (Chen et al., 2024a; Zhang et al., 2024b;a; Wang et al., 2024a) have focused on accelerating the inference of MLLMs by reducing the number of visual tokens while preserving essential information. For instance, FasterVLM (Zhang et al., 2024a) and VisionZip (Yang et al., 2024) perform global dominant visual token selection after vision encoding, while FastV (Chen et al., 2024a) and SparseVLM (Zhang et al., 2024b) prune tokens based on attention weights during LLM decoding. Although these methods achieve promising performance, they predominantly focus on individual components of the MLLM framework. Furthermore, existing approaches overlook the necessity of aligning visual token selection with textual information. This oversight results in loss of textual context, which is

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

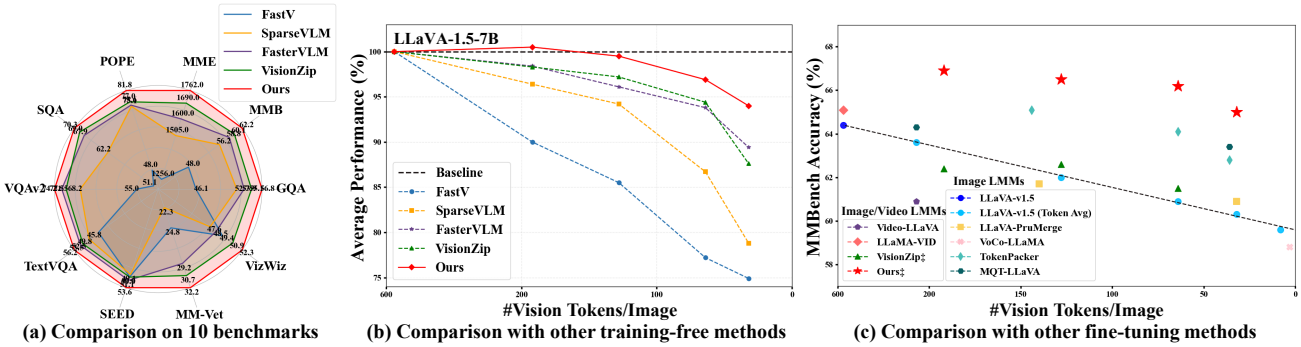


Figure 2. **Performance comparisons with other methods.** (a) Comparison on LLaVA-1.5-7B across 10 benchmarks, with a **88.9%** reduction ratio. (b)&(c) Relationship between performance and efficiency of different methods on LLaVA-1.5-7B, in both training-free and fine-tuning scenarios, respectively. Our VisionTrim‡ only uses 1/10 of the original LLaVA-1.5 dataset for efficient fine-tuning.

essential for accurate LLM decoding, ultimately leading to a substantial degradation in performance.

To address these issues, we propose a unified vision token compression framework, named **VisionTrim** for training-free acceleration of MLLMs. As shown in Figure 1, unlike previous methods that focus solely on visual token compression during either vision encoding or LLM decoding, our approach considers the entire forward propagation of the MLLM. We introduce two plug-and-play modules that accelerate both vision encoding and LLM decoding processes, which can be seamlessly inserted between any two layers of the vision encoder and the LLM. Specifically, our proposed method primarily consists of two key components: Dominant Vision Token Selection (DVTS) and Text-Guided Vision Complement (TGVC) modules.

First, in DVTS module, we consider both global semantics and local spatial continuity to filter visual tokens that carry essential visual information. This ensures that critical visual details are retained while reducing redundancy. Second, in TGVC module, we leverage textual information to guide the clustering and merging of the pruned visual tokens that are relevant to the input text instructions. These tokens are then used to complement the dominant visual tokens from DVTS module. By integrating textual context into the visual token reduction process, our approach enhances the implicit alignment between visual and textual representations, thereby improving the overall efficiency of the pruning MLLM. As shown in Figure 2, our approach consistently outperforms previous methods across various reduction ratios for both image- and video-based MLLMs. In summary, the contributions of our work are threefold:

- We propose a unified and comprehensive vision token compression framework, named VisionTrim for training-free MLLM acceleration.
- Two novel plug-and-play modules, DVTS and TGVC, are presented to accelerate both the vision encoder and LLM forward processes.
- Extensive experiments on various multimodal benchmarks for both image- and video-based MLLMs demonstrate the superiority of our VisionTrim.

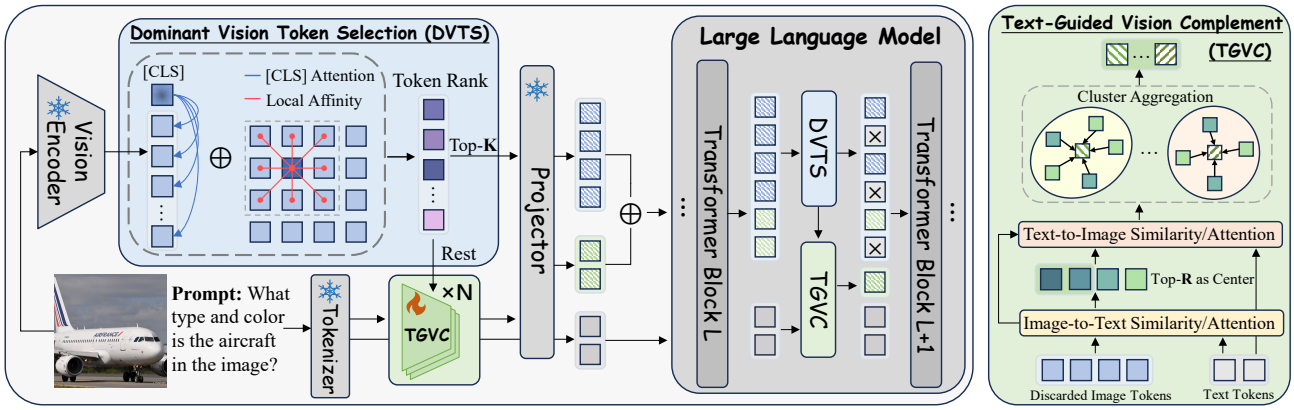
2. Related Work

2.1. Multimodal Large Language Models

Large Language Models (LLMs) (Vicuna, 2023; Touvron et al., 2023; Bai et al., 2023a; Achiam et al., 2023) have garnered significant attention due to their powerful capabilities in natural language processing tasks such as text understanding, generation, and question answering. Nonetheless, the reliance on purely textual data limits their applicability, as human perception is inherently multimodal. This has spurred the development of Multimodal LLMs (MLLMs) (Liu et al., 2023; Bai et al., 2023b; Chen et al., 2024b; Reid et al., 2024), which integrate LLMs with visual encoders to augment performance in multimodal tasks. The typical image- and video-based MLLMs (Liu et al., 2024b;c; Cheng et al., 2024; Lin et al., 2023) utilizes an MLP to project visual information encoded by a Vision Transformer (ViT) (Dosovitskiy, 2020) into a space interpretable by LLMs, improving performance on visual-language tasks through visual instruction tuning. However, this paradigm requires a large number of visual tokens to represent visual information, particularly with high-resolution images and video inputs, which further exacerbates the issue. The resulting increase in computational demands and inference times poses significant challenges, hindering the practical deployment of MLLMs in real-world applications.

2.2. Token Compression for MLLMs

The quadratic complexity inherent in Transformer networks (Vaswani, 2017), which scales with the sequence length of input tokens, remains a widely acknowledged challenge. To address this issue, several methods (Li et al., 2023a; Bai et al., 2023b; Cha et al., 2024; Li et al., 2024b; Yao et al., 2024; Hu et al., 2024) have explored more ef-



(a) Overall training-free framework **(b) Structure of TGVC module**

Figure 3. (a) Overview of our proposed VisionTrim with the detailed DVTS module, and (b) structure of the TGVC module. Both DVTS and TGVC modules can be generally utilized in both the vision encoding stage and the LLM decoding stage.

ficient visual projectors that enable compact visual representations using fewer visual tokens before feeding them into the LLM. While these approaches have demonstrated promising performance, they often necessitate architectural modifications and extensive model training, resulting in significant computational costs. In contrast, other recent works (Shang et al., 2024; Chen et al., 2024a; Zhang et al., 2024a;b; Yang et al., 2024) have focused on reducing visual token counts in a training-free manner. FastV (Chen et al., 2024a) prunes redundant tokens at a specific layer of LLM based on attention scores. LLaVA-PruMerge (Shang et al., 2024) leverages class-spatial similarity for pruning and merging visual tokens. FasterVLM (Zhang et al., 2024a) further evaluates token importance by examining the attention scores between the `[CLS]` token and image tokens. SparseVLM (Zhang et al., 2024b) relies on text-visual attention within LLM and recycles the pruned tokens, while VisionZip (Yang et al., 2024) aggregates substantial information and merges retained tokens in a text-agnostic manner. Unlike these methods, our approach simultaneously integrates both global semantic significance and local spatial continuity to preserve visual integrity. More importantly, we introduce a text-guided visual complement mechanism to ensure alignment with textual instructions, offering a more comprehensive and effective solution to the challenge of visual token reduction.

3. Methodology

Unlike previous methods that focus solely on token compression within the vision encoder or during the forward propagation of LLM, our approach comprehensively considers the entire processing pipeline of MLLM. We introduce two plug-and-play modules designed to simultaneously accelerate both the vision encoder and LLM forward processes. These modules can be seamlessly integrated between any two layers of either the vision encoder or the LLM.

Our approach primarily consists of two key components: The first component filters tokens to retain those that carry essential **visual information**, paying particular attention to their importance for both global semantics and local spatial continuity. The second component employs **textual information** to guide the clustering and merging of the retained visual tokens that are pertinent to the input text instructions. This process helps complement the dominant visual tokens that contain critical visual details. Figure 3 illustrates the whole architecture of our approach.

3.1. Domiant Vision Token Selection

To preserve visual integrity during visual token compression, we introduce a visual token importance scoring mechanism for dominant vision token selection. This mechanism comprehensively incorporates both global semantic significance and local spatial continuity. Specifically, we first utilize `[CLS]` token's attention scores relative to other visual tokens of the CLIP-based vision encoder as the criterion to assess global semantic importance. Next, we develop a Local Token Affinity Measure (LTAM) algorithm, which employs a dual-kernel to capture both feature similarity and spatial proximity, ensuring local spatial continuity. These complementary metrics are then integrated using an adaptive variance-based weighting scheme to prioritize the selection of more reliable visual tokens.

Global Semantic Importance. Motivated by previous methods (Yang et al., 2024; Zhang et al., 2024a), the `[CLS]` token's attention distribution over all image tokens serves as a natural measure of the global semantic significance. We extract the attention weights from the penultimate layer of the CLIP-based vision encoder and leverage the attention patterns from the `[CLS]` token. The self-attention computation for the `[CLS]` token follows:

$$\mathbf{Q}_{[\text{CLS}]} = \mathbf{W}_{\mathbf{Q}} X_{[\text{CLS}]}^{L-1}, \quad \mathbf{K}_i = \mathbf{W}_{\mathbf{K}} X_i^{L-1}, \tag{1}$$

$$A^{L-1}_{[\text{CLS}],i} = \text{softmax}(\mathbf{Q}_{[\text{CLS}]}\mathbf{K}_i^T/\sqrt{d_k}), \quad i \in [1, N]. \quad (2)$$

Here, $X^{L-1}_{[\text{CLS}]}$ and X^{L-1}_i denote the hidden states of the `[CLS]` token and the i-th visual token at the $(L-1)$-th layer. $\mathbf{W_Q}$ and $\mathbf{W_K}$ are learnable projection matrices, and d_k is the key vector dimension. N is the total number of visual tokens. The global importance score S^g_i for the i-th visual token is the average attention score across all heads:

$$S^g_i = \frac{1}{H}\sum_{h=1}^{H} A^{L-1}_{[\text{CLS}],i,h}, \quad i \in [1, N]. \quad (3)$$

This formulation effectively measures each visual token's contribution to the global semantic representation of the image based on the `[CLS]` token's attention mechanism. The computed global scores $\{S^g_i\}^N_{i=1}$ are normalized to yield a probability distribution over all visual tokens, *i.e.* $\hat{S}^g_i = \exp(S^g_i)/\sum_{j=1}^{N}\exp(S^g_j)$.

Local Spatial Continuity. To effectively capture the local spatial continuity of visual tokens, we introduce the Local Token Affinity Measure (LTAM) algorithm, which employs a dual-kernel affinity mechanism that simultaneously accounts for both feature similarity and positional proximity. For the i-th token at position (x, y), its local importance S^l_i is determined by computing the affinity with neighboring tokens within a local kernel $\mathcal{N}(x, y)$ of size $k \times k$. For tokens positioned at (x, y) and (u, v), the affinity kernel κ^* is defined as a weighted combination of a feature-based term κ_{feat} and a position-based term κ_{pos}:

$$\kappa^{xy,uv}_{feat} = -\left(\frac{\|F_{xy}-F_{uv}\|}{w_1\sigma_f}\right)^2, \kappa^{xy,uv}_{pos} = -\left(\frac{\|P_{xy}-P_{uv}\|}{w_2\sigma_p}\right)^2, \quad (4)$$

$$\kappa^{*xy,uv} = \frac{\exp(\kappa^{xy,uv}_{feat})}{\sum_{(h,w)}\exp(\kappa^{xy,hw}_{feat})} + w_3\frac{\exp(\kappa^{xy,uv}_{pos})}{\sum_{(h,w)}\exp(\kappa^{xy,hw}_{pos})}, \quad (5)$$

where $F_{xy} \in \mathbb{R}^d$ and $P_{xy} \in \mathbb{R}^2$ denote the feature vector and spatial coordinates of the token at (x, y), respectively, while σ_f and σ_p represent the standard deviations of the feature and positional differences, respectively. The pair (h, w) is sampled from the neighborhood set of $\mathcal{N}(x, y)$, and w_1, w_2, and w_3 are balancing parameters. The local importance S^l_i of the i-th token at position (x, y) is then computed by averaging the affinity scores κ^* over all neighboring tokens.

Adaptive Variance-based Weighting. To integrate global and local importance scores, we propose an adaptive variance-based weighting mechanism:

$$S_i = \alpha\hat{S}^g_i + (1-\alpha)S^l_i, \text{ where } \alpha = \frac{\sigma^2_l}{\sigma^2_g + \sigma^2_l}. \quad (6)$$

σ^2_g and σ^2_l denote the variances of the global and local importance scores, respectively. This adaptive weighting scheme automatically prioritizes more reliable signals based on their consistency, ensuring robust token selection. The final importance scores, $\{S_i\}^N_{i=1}$, are used to select the top-K informative tokens $\mathbf{V}_{dom} \in \mathbb{R}^{K\times d}$ from the complete set $\mathbf{V} \in \mathbb{R}^{N\times d}$. This selection process ensures the preservation of both semantic relevance and spatial continuity.

3.2. Text-Guided Vision Complement

Although the selected tokens capture primary visual information, we do not account for their relevance to the input text instruction. In other words, the selected visual tokens and textual information are not fully aligned, potentially leading to the loss of crucial visual tokens associated with the textual instructions. To address this, we utilize the text instruction to guide the selection of relevant visual tokens. Leveraging CLIP's text encoder, we calculate the similarity between the remaining visual tokens and the text tokens, identifying the top R tokens as clustering centers. These centers serve as references for assigning the remaining visual tokens to the R clusters. We then merge each cluster to derive the final R visual tokens most relevant to the text, which we refer to as the vision complement tokens.

Determining Clustering Centers. Given the remaining visual tokens $\mathbf{V}_r \in \mathbb{R}^{(N-K)\times d}$ after dominant token selection, we first calculate their similarity $S_{t2v} \in \mathbb{R}^{L\times(N-K)}$ with the text features $T \in \mathbb{R}^{L\times d}$ to identify potential clustering centers:

$$S_{t2v} = \text{softmax}(\frac{T\mathbf{V}_r^T}{\sqrt{d}}). \quad (7)$$

Next, the token-level importance scores $s \in \mathbb{R}^{N-K}$ are determined by averaging the similarity scores across all text tokens, given by $s = \frac{1}{L}\sum_{i=1}^{L} S_{t2v_i}$. The top-$R$ tokens are then selected as clustering centers, denoted as $C = \{c_1, ..., c_R\}$.

Token Assignment. For each remaining token $v_i \in \mathbf{V}_r \setminus C$, we compute its assignment score to each clustering center using text-guided similarity. Specifically, for a center c_j, the similarity scores are calculated as follows:

$$S^i_{v2t} = \text{softmax}(\frac{v_iT^T}{\sqrt{d}}), \quad S^j_{t2c} = \text{softmax}(\frac{Tc_j^T}{\sqrt{d}}). \quad (8)$$

The assignment score a_{ij} is then given by

$$a_{ij} = S^i_{v2t}S^j_{t2c}. \quad (9)$$

Each token is assigned to the clustering center with the highest similarity score:

$$\text{cluster}(v_i) = \arg\max_j a_{ij}. \quad (10)$$

Cluster Aggregation. For each cluster centered at c_j, we aggregate the assigned tokens through weighted averaging

based on their text-guided similarities:

$$v_j^{\text{com}} = c_j + \sum_{v_i \in \text{cluster}(j)} \frac{a_{ij}}{\sum_{v_k \in \text{cluster}(j)} a_{kj}} v_i. \quad (11)$$

This process is repeated for T iterations to refine the clusters. The final vision complement tokens $\mathbf{V}_{com} = \{v_1^{com}, v_2^{com}, ...v_R^{com}\}$ are then concatenated with the dominant tokens to form the complete visual representation:

$$\mathbf{V}_{final} = [\mathbf{V}_{dom}; \mathbf{V}_{com}] \in \mathbb{R}^{(K+R)\times d}. \quad (12)$$

This text-guided complement mechanism ensures that visual tokens capture key patterns while aligning with textual instruction, enhancing multimodal reasoning in the subsequent LLM propagation stage.

3.3. Multi-Stage Pruning Strategy

Our Dominant Vision Token Selection (DVTS) and Text-Guided Vision Token Complement (TGVC) modules provide a versatile approach to token reduction that can be effectively applied at two stages of the MLLM pipeline.

1) **Vision Encoding Stage**: Before LLM processing, DVTS and TGVC reduce the initial visual token sequence $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N\}$ to a more compact representation $\mathbf{V}' = \{\mathbf{v}'_1, \mathbf{v}'_2, ..., \mathbf{v}'_{K+R}\}$, where $K + R < N$.

2) **LLM Decoding Stage**: DVTS and TGVC can be integrated between transformer layers during LLM decoding, enabling dynamic token pruning while keeping cross-modal alignment intact. At layer l, cross-attention scores between visual and textual tokens are computed as follows:

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{H}_v^l \mathbf{H}_t^l}{\sqrt{D}}) \in \mathbb{R}^{N_v \times N_t}, \quad \alpha_i = \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbf{A}_{i,j}, \quad (13)$$

where $\mathbf{H}_v^l \in \mathbb{R}^{N_v \times D}$ and $\mathbf{H}_t^l \in \mathbb{R}^{N_t \times D}$ represent visual and textual tokens at layer l, and α_i denotes the average attention score for the i-th visual token. Using these scores and local spatial affinity scores from the LTAM scheme, we select the top-M tokens and perform token complement:

$$\mathbf{V}_{final} = \text{TopM}(\{\mathbf{H}_v^l[i], \mathbf{S}_i^l\}_{i=1}^{N_v}; \alpha, M) + \sum_{j \in \mathcal{U}(\alpha)} w_j \mathbf{H}_v^l[j]. \quad (14)$$

Here, $\text{TopM}(\cdot; \alpha, M)$ selects M tokens with the highest dual-attention scores α, $\mathcal{U}(\alpha)$ denotes the set of unselected tokens, and w_j represents the learned aggregation weights. The multi-stage application of DVTS and TGVC refines visual representation while ensuring both computational efficiency and effective cross-modal alignment.

4. Experiments

4.1. Experimental Setting

Datasets and Benchmarks. We conduct an extensive evaluation across 10 widely-used image-based benchmarks to assess the multi-modal understanding and reasoning capability of our proposed model, including common visual question answering tasks, like GQA (Hudson & Manning, 2019), VQA^{V2} (Goyal et al., 2017) and VizWiz (Gurari et al., 2018), as well as other multi-modal benchmarks such as POPE (Li et al., 2023c), MMBench (Liu et al., 2025), MME (Fu et al., 2023) and MM-Vet (Yu et al., 2023). Additionally, we also experiment on 4 widely used video-based multi-modal understanding tasks, including TGIF-QA (Jang et al., 2017), MSVD-QA (Xu et al., 2017), MSRVTT-QA (Xu et al., 2017) and ActivityNet-QA (Yu et al., 2019).

Implementaion Details. We apply our method to various open-source MLLMs, including the classic LLaVA-1.5 (Liu et al., 2024b) model for normal-resolution images, LLaVA-NeXT (Liu et al., 2024c) for high-resolution images, Video-LLaVA (Lin et al., 2023) for video-based tasks, and LLaVA-OneVision (Li et al., 2024a) and QWen2-VL (Wang et al., 2024b) for broader validation. To ensure a fair comparison, we adopt the default settings and evaluation metrics as reported in their respective papers. We compare our approach with FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2024b), FasterVLM (Zhang et al., 2024a), and VisionZip (Yang et al., 2024). Unlike these methods, our approach is applicable at multiple stages of the MLLM pipeline, facilitating more effective cross-modal alignment.

4.2. Main Results

Normal Resolution. As shown in Table 1, we first evaluate our approach on LLaVA-1.5-7B under the normal-resolution setting. Even with only 32 tokens retained, VisionTrim outperforms the existing methods with 64 tokens on most metrics, demonstrating the effectiveness of our unified token compression strategy. Moreover, VisionTrim consistently surpasses previous methods across all configurations on token numbers (192, 128, and 64). In benchmarks, such as POPE, SQA, and TextVQA, VisionTrim not only maintains its performance without degradation but also achieves improvements, highlighting the severe redundancy in visual tokens fed to the LLM.

High Resolution. Our approach minimizes token count while maintaining performance on LLaVA-Next-7B with high-resolution inputs. As shown in Table 2, VisionTrim retains 99.8% of the original performance using only 77.8% tokens, without extra training. With nearly 95% token reduction, it achieves 93.8% performance without training and 96.7% after fine-tuning, surpassing the previous SOTA method VisionZip by 3.3% and 1.2%, respectively. This validates VisionTrim's effectiveness for high-resolution inputs.

Table 1. Comparison with other methods on LLaVA-1.5-7B. The vanilla number of visual tokens is 576. The first line of each method shows the raw benchmark accuracy, while the second line indicates the proportion relative to the upper limit. The last column presents the average value. Ours‡ refers to fine-tuning the cross-modality attention module and projector using 1/10 of the original LLaVA-1.5 dataset.

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	SEED	MMVet	VizWiz	Avg.
Upper Bound, 576 Tokens (**100%**)											
Vanilla (Liu et al., 2024a)	61.9 100%	64.7 100%	1862 100%	85.9 100%	69.5 100%	78.5 100%	58.2 100%	58.6 100%	31.1 100%	50.1 100%	100%
Retain Averaged 192 Tokens (↓ 66.7%)											
FastV (Chen et al., 2024a)	52.7 85.1%	61.2 94.6%	1612 86.6%	64.8 75.4%	67.3 96.8%	67.1 85.5%	52.5 90.2%	57.1 97.4%	27.7 89.1%	49.5 98.8%	90.0%
SparseVLM (Zhang et al., 2024b)	57.6 93.1%	62.5 96.6%	1721 92.4%	83.6 97.3%	69.1 99.4%	75.6 96.3%	56.1 96.4%	55.8 95.2%	30.2 97.1%	50.0 99.8%	96.4%
VisionZip (Yang et al., 2024)	59.3 95.8%	63.0 97.4%	1783 95.8%	85.3 99.3%	68.9 99.1%	76.8 97.8%	57.3 98.5%	56.4 96.2%	31.7 101.9%	50.5 100.8%	98.3%
Ours	60.5 97.7%	64.4 99.5%	1796 96.5%	86.8 101.0%	70.8 101.9%	78.0 99.4%	58.4 100.3%	58.3 99.5%	33.2 106.8%	51.3 102.4%	**100.5%** ↑ (2.2%)
Ours‡	62.0 100.2%	66.9 103.4%	1822 97.9%	87.6 102.0%	70.4 101.3%	78.6 100.1%	58.9 101.2%	59.0 100.7%	32.8 105.5%	51.9 103.6%	101.6%
Retain Averaged 128 Tokens (↓ 77.8%)											
FastV (Chen et al., 2024a)	49.6 80.1%	56.1 86.7%	1490 80.0%	59.6 69.4%	60.2 86.6%	61.8 78.7%	50.6 86.9%	55.9 95.4%	27.7 89.1%	50.9 101.6%	85.5%
SparseVLM (Zhang et al., 2024b)	56.0 90.5%	60.0 92.7%	1696 91.1%	80.5 93.7%	67.1 96.5%	73.8 94.0%	54.9 94.3%	53.4 91.1%	30.0 96.5%	51.0 101.8%	94.2%
VisionZip (Yang et al., 2024)	57.6 93.1%	62.0 95.8%	1762 94.6%	83.2 96.9%	68.9 99.1%	75.6 96.3%	56.8 97.6%	54.9 93.7%	32.6 104.8%	50.0 99.8%	97.2%
Ours	59.3 95.8%	63.8 98.6%	1788 96.0%	85.6 99.7%	69.7 100.3%	77.2 98.3%	58.2 100.0%	57.5 98.1%	32.7 105.1%	51.4 102.6%	**99.5%** ↑ (2.3%)
Ours‡	61.6 99.5%	66.5 102.8%	1847 99.2%	86.1 100.2%	69.8 100.4%	78.0 99.4%	58.5 100.5%	58.6 100.0%	32.0 102.9%	52.0 103.8%	100.9%
Retain Averaged 64 Tokens (↓ 88.9%)											
FastV (Chen et al., 2024a)	46.1 74.5%	48.0 74.2%	1256 67.5%	48.0 55.9%	51.1 73.5%	55.0 70.1%	47.8 82.1%	51.9 88.6%	25.8 83.0%	51.4 102.6%	77.2%
SparseVLM (Zhang et al., 2024b)	52.7 85.1%	56.2 86.9%	1505 80.8%	75.1 87.4%	62.2 89.5%	68.2 86.9%	51.8 89.0%	51.1 87.2%	23.3 74.9%	49.6 99.0%	86.7%
VisionZip (Yang et al., 2024)	55.1 89.0%	60.1 92.9%	1690 90.8%	77.0 89.6%	69.0 99.3%	72.4 92.2%	55.5 95.4%	52.2 89.1%	31.7 101.9%	51.9 103.6%	94.4%
Ours	56.8 91.8%	62.2 96.1%	1762 94.6%	81.8 95.2%	70.3 101.2%	74.2 94.5%	56.2 96.6%	53.6 91.5%	32.2 103.5%	52.3 104.4%	**96.9%** ↑ (2.5%)
Ours‡	61.0 98.5%	66.2 102.3%	1875 100.7%	85.5 99.5%	71.4 102.7%	77.4 98.6%	57.3 98.5%	55.8 95.2%	32.8 105.5%	52.8 105.4%	100.7%
Retain Averaged 32 Tokens (↓ 94.4%)											
Ours	55.3 89.3%	60.2 93.0%	1646 88.4%	77.5 90.2%	70.1 100.9%	72.6 92.5%	54.7 94.0%	52.5 89.6%	30.4 97.7%	52.2 104.2%	94.0%
Ours‡	59.8 96.6%	65.0 100.5%	1766 94.8%	84.1 97.9%	69.2 99.6%	75.7 96.4%	55.8 95.9%	55.5 94.7%	32.1 103.2%	52.3 104.4%	98.4%

Video-based Multimodal Understanding. To assess the generalization of our method across modalities, we apply it to Video-LLaVA-7B, which processes 8 frames from a video and generates 2048 visual tokens. Following the baseline of SparseVLM, we prune visual tokens down to 136. As shown in Table 3, VisionTrim achieves 98.0% of the original performance with a 93.4% pruning ratio, outperforming all other methods on four benchmarks. Furthermore, VisionTrim consistently exceeds 96.0% performance, demonstrating its effectiveness and strong robustness. Our method excels even with high pruning ratios, balancing inference speed and accuracy for video tasks with temporal features.

Broader Validation. To further evaluate the effectiveness of VisionTrim, we deploy it on the state-of-the-art open-source MLLMs, LLaVA-OneVision-7B and QWen2-VL-7B, using approximately 1/3 of the original input tokens. As shown in Table 6, VisionTrim demonstrates competitive performance with only around 0.1%-0.4% performance loss in most cases and occasionally outperforms the baseline MLLM. Notably, VisionTrim exceeds the vanilla QWen2-VL by 0.5% on the MMB dataset and 0.2% on the MV-Bench dataset, confirming its effectiveness in reducing visual redundancy.

4.3. Ablation Study

Dual-Attention Mechanism in DVTS. We employ various ensemble strategies to combine global semantic information from the `[CLS]` token attention and local spatial affinity captured by our proposed LTAM algorithm in the DVTS module, as shown in Table 4. Specifically, we explore three ensemble methods: element-wise maximum,

Table 2. Comparison with other methods on LLaVA-NeXT-7B.

Method	GQA	MMB	MME	SQA	VQA^{V2}	VQA^{T}	POPE	Avg.
Upper Bound, 2880 Tokens (**100%**)								
Vanilla	64.2	67.9	1842	70.2	80.1	61.3	86.3	100%
	100%	100%	100%	100%	100%	100%	100%	
Retain 640 Tokens (↓ 77.8%)								
SparseVLM	60.3	65.7	1772	67.7	77.1	57.8	85.2	96.1%
	93.9%	96.8%	96.2%	96.4%	96.3%	94.3%	98.7%	
VisionZip	61.3	66.3	1787	68.1	79.1	60.2	87.7	98.0%
	95.5%	97.6%	97.0%	97.0%	98.8%	98.2%	101.6%	
Ours	63.2	67.2	1825	70.7	79.8	61.0	88.5	**99.8%**
	98.4%	99.0%	99.1%	100.7%	99.6%	99.5%	102.5%	↑ (1.8%)
Ours‡	64.4	67.6	1853	72.8	80.2	61.5	88.2	101.0%
	100.3%	99.6%	100.6%	103.7%	100.1%	100.3%	102.2%	
Retain 320 Tokens (↓ 88.9%)								
SparseVLM	57.7	64.3	1694	67.3	73.4	55.9	78.6	92.3%
	89.9%	94.7%	92.0%	95.9%	91.6%	91.2%	91.1%	
VisionZip	59.3	63.1	1702	67.3	76.2	58.9	82.1	94.3%
	92.4%	92.9%	92.4%	95.9%	95.1%	96.1%	95.1%	
Ours	61.7	64.8	1795	69.6	77.3	59.6	83.6	**97.0%**
	96.1%	95.4%	97.4%	99.1%	96.5%	97.2%	96.9%	↑ (2.7%)
Ours‡	62.5	66.4	1816	71.5	78.9	61.2	85.1	98.9%
	97.4%	97.8%	98.6%	101.9%	98.5%	99.8%	98.6%	
Retain 160 Tokens (↓ 94.4%)								
SparseVLM	51.2	63.1	1542	67.5	66.3	46.4	77.3	85.8%
	79.8%	92.9%	83.7%	96.2%	82.8%	75.7%	89.6%	
VisionZip	55.5	60.1	1630	68.3	71.4	56.2	79.4	90.5%
	86.4%	88.5%	88.5%	97.3%	89.1%	91.7%	92.0%	
Ours	57.2	63.3	1702	70.2	74.2	58.3	81.1	**93.8%**
	89.1%	93.2%	92.4%	100.0%	92.6%	95.1%	94.0%	↑ (3.3%)
Ours‡	59.7	63.8	1768	71.6	77.5	60.5	83.5	96.7%
	93.0%	94.0%	96.0%	102.0%	96.8%	98.7%	96.8%	

geometric mean, and adaptive variance-based weighting. Compared to the baseline, which solely uses [CLS] token attention, incorporating both global semantic and local spatial continuity yields significant performance improvements. Furthermore, as depicted in Figure 4, relying exclusively on [CLS] token results in the loss of crucial semantic information, while considering local spatial continuity helps retain better visual token coverage. Consequently, our proposed dual-attention filtering mechanism offers a more holistic approach to attention integration.

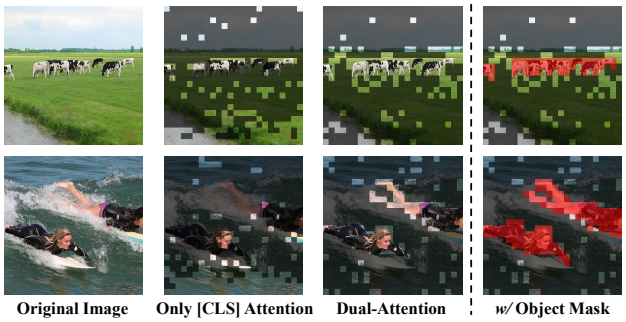


Figure 4. Visualization of retained visual patches with and without dual-attention mechanism in DVTS module. The areas masked in black represent the discarded visual tokens.

Table 3. Comparison with other methods on Video-LLaVA-7B. The original number of video tokens is 2048, while in our experiment this is pruned to only 136 tokens.

Method	TGIF		MSVD		MSRVTT		ActivityNet		Avg.	
	Acc	Score	Acc	Score	Acc	Score	Acc	Score	Acc	Score
Vanilla	47.1	3.35	69.8	3.92	56.7	3.48	43.1	3.35	100.0%	+0.00
FastV	23.1	2.47	38.0	2.71	19.3	2.02	30.6	2.82	52.1%	-1.02
	49.0%	-0.88	54.4%	-1.21	34.0%	-1.46	71.0%	-0.53		
SparseVLM	44.7	3.29	68.2	3.90	31.0	2.68	42.6	3.32	86.5%	-0.23
	94.9%	-0.06	97.7%	-0.02	54.7%	-0.80	98.8%	-0.03		
VisionZip	42.0	3.16	63.5	3.58	49.6	3.34	42.0	3.21	91.3%	-0.20
	89.2%	-0.19	91.0%	-0.34	87.5%	-0.14	97.4%	-0.14		
Ours	45.2	3.32	68.6	3.93	54.9	3.42	43.5	3.35	**98.0%**	**-0.02**
	96.0%	-0.03	98.3%	0.01	96.8%	-0.06	100.9%	0.00	↑ (6.7%)	↑ (0.18)

Table 4. Ablation study of various ensemble strategies for [CLS] attention and local affinity scores in DVTS module. Here we use 64 tokens on LLaVA-1.5-7B.

Ensemble Strategy	GQA	MMB	MME	POPE	SQA	VQA^{Text}
Only [CLS] token	52.8	55.3	1536	74.2	67.9	51.2
Maximum	53.4	55.7	1654	75.6	68.9	52.4
Geometric Mean	54.2	56.5	1631	75.2	**70.0**	52.3
Adaptive Weighting	**54.7**	**56.9**	**1656**	**76.1**	69.9	**52.6**

Table 5. Ablation study on TGVC module. This experiment, conducted before inputting data into the LLM, evaluates the effectiveness of the TGVC in reducing noise within text-visual attention during the LLM's forward pass.

Benchmark	Number of Tokens				Avg.
	192	128	64	32	
POPE	83.0	81.4	76.1	72.9	78.4
w/ TGVC	86.1 (↑ 3.1)	84.7 (↑ 3.3)	80.2 (↑ 4.1)	77.3 (↑ 4.4)	82.1 (↑ 3.7)
MMBench	61.1	60.5	56.9	54.9	58.4
w/ TGVC	63.4 (↑ 2.3)	62.9 (↑ 2.4)	60.2 (↑ 3.3)	59.1 (↑ 4.2)	61.4 (↑ 3.0)
Text-VQA	55.3	54.4	52.6	50.2	53.1
w/ TGVC	57.8 (↑ 2.5)	57.2 (↑ 2.8)	56.0 (↑ 3.4)	54.2 (↑ 4.0)	56.3 (↑ 3.2)

Information Complement of TGVC. As shown in Table 5, incorporating the TGVC module significantly boosts performance across three multimodal tasks: POPE, MMBench, and Text-VQA. Notably, as the compression ratio increases and token count decreases, the TGVC module's impact becomes more pronounced, resulting in performance gains exceeding 4%. Additionally, Figure 5 illustrates that the TGVC module retains essential visual tokens related to textual instructions, ensuring critical visual information is not pruned away. It can also be applied multiple times for enhanced visual completion, allowing textual tokens to better align with the corresponding visual information in the subsequent LLM decoding stage.

Stage-Aware Pruning Strategy. We conduct a thorough ablation study to evaluate our multi-stage vision pruning design (Table 7), reducing image tokens to 64 for an 88.9% reduction. Initially, applying DVTS and TGVC modules solely in the vision encoder improves multimodal processing and reduces KV cache memory by 91.6%. For a fair comparison, we also implement the text-guided clustering

Table 6. Experiment results of deploying VisionTrim on LLaVA-OneVision-7B and QWen2-VL-7B over single-/multi-image and video benchmarks. For LLaVA-OneVision, we utilize 243 tokens per image (down from the original 729 tokens) and 66 tokens per video frame (reduced from the original 196 tokens). Similarly, for QWen2-VL, approximately 1/3 of the original input tokens are used.

Model	Single-image Benchmarks				Multi-image Benchmarks			Video-based Benchmarks			
	MMMU	MMStar	MMB	MMVet	MuirBench	Mantis	Q-Bench	Video-MME$_{(wo/w\ subs)}$	PerceptionTest	Egoschema	MV-Bench
LLaVA-OneVision	48.8	61.7	80.8	57.5	41.8	64.2	74.4	58.2 / 61.5	57.1	60.1	56.7
w/ VisionTrim	48.6 (↓ 0.2)	60.2 (↓ 1.5)	80.5 (↓ 0.3)	57.6 (↑ 0.1)	41.4 (↓ 0.4)	64.4 (↑ 0.2)	74.3 (↓ 0.1)	58.4 (↑ 0.2) / 61.4 (↓ 0.1)	57.0 (↓ 0.1)	60.2 (↑ 0.1)	56.5 (↓ 0.2)
QWen2-VL	54.1	60.7	80.7	62.0	–	–	–	63.3 / 69.0	62.3	66.7	67.0
w/ VisionTrim	53.8 (↓ 0.3)	59.6 (↓ 1.1)	81.2 (↑ 0.5)	61.8 (↓ 0.2)	–	–	–	63.1 (↓ 0.2) / 68.9 (↓ 0.1)	62.0 (↓ 0.3)	66.5 (↓ 0.2)	67.2 (↑ 0.2)

Table 7. Ablation study of different pruning strategies at the vision encoding and LLM decoding stages.

Stages	Tokens	GQA	MMB	POPE	VQA^{V2}	KV Cache Memory (MB)
LLaVA-1.5-7B	576	61.9	64.7	85.9	78.5	303.6
Only in ViT *w/* DVTS+TGVC	64	55.6	60.2	80.2	72.2	**25.4** (↓ 91.6%)
Only in LLM *w/* TGVC	64	54.7	58.9	76.1	72.3	43.5 (↓ 85.7%)
Both in ViT and LLM	64	**56.8**	**62.2**	**81.8**	**74.2**	30.2 (↓ 90.1%)

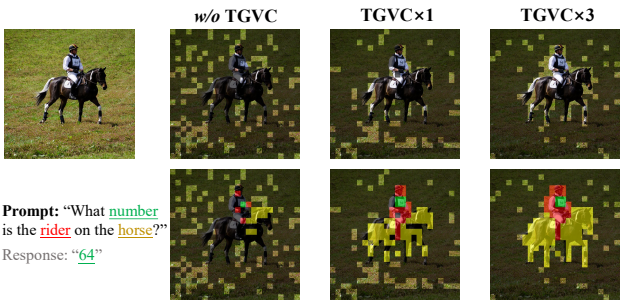


Figure 5. Visualization of retained visual patches with and without TGVC module. We show the correspondence between the salient visual regions and text by different colors.

and merging only in LLM's decoding stage, yielding performance gains of 4.1% and 2.7% over SparseVLM on VQA^{V2} and MMB datasets, respectively. When applied to both vision encoding and LLM decoding stages, VisionTrim significantly outperforms existing SOTA methods and reduces memory usage by 90.1%. Moreover, Figure 6 shows attention maps with and without VisionTrim, highlighting that the vanilla LLM exhibits high redundancy and poor cross-modal alignment. In contrast, VisionTrim improves cross-modal alignment and reduces visual redundancy without compromising performance. *Please refer to the Appendix B for more case studies and visualization results.*

4.4. Efficiency Analysis

We evaluate the efficiency of our method in terms of CUDA time, FLOPs, and storage memory, and compare it with vanilla LLaVA-NeXT-7B and other methods, as shown in Table 8. With an 88.9% reduction ratio, our method reduces CUDA time by 53.2%, FLOPs by 88.5%, and storage memory by 90.4%, while maintaining 97.2% accuracy on VQA^{Text}. Notably, when retaining the same token count, our method is 22.6% faster than SparseVLM in inference

Table 8. Efficiency analysis of our method on LLaVA-NeXT-7B. The detailed metric includes latency (CUDA time), computation (FLOPs), and storage (cache memory).

Methods	Tokens	VQA^{Text} (%) ↑	CUDA Time (Min & Sec) ↓	△	FLOPs (T) ↓	△	KV Cache (MB) ↓	△
Vanilla	2880	61.3	26:34	–	9.6	–	1512.1	–
SparseVLM	320	55.9	18:26	30.6%	1.5	84.4%	168.0	88.9%
VisionZip	320	58.9	17:53	32.7%	1.6	83.3%	180.4	88.1%
Ours	320	**59.6**	**12:26**	53.2%	**1.1**	88.5%	**144.5**	90.4%

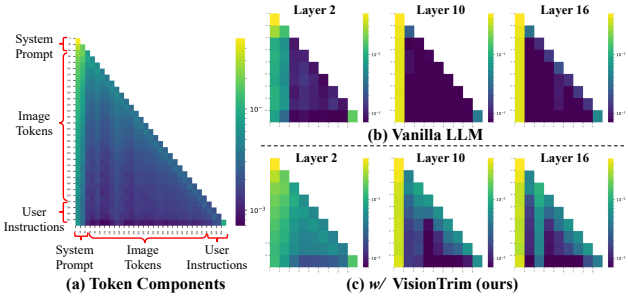


Figure 6. Comparison of attention maps during LLM processing with and without our proposed VisionTrim. After processing through DVTS and TGVC modules, MLLM improves cross-modal alignment while significantly reducing redundant visual tokens.

time and requires 5.2% less computational budget than VisionZip, while minimizing KV cache memory usage. These results underscore the high efficiency of our approach.

5. Conclusion and Limitation

In this paper, we proposed VisionTrim, a unified training-free framework for MLLM acceleration through comprehensive vision token compression. We presented two plug-and-play modules that accelerated both vision encoding and LLM decoding stages. By integrating the DVTS module, which selects tokens based on global semantics and local spatial continuity, with the TGVC module, which performs text-guided token clustering and aggregation, our approach consistently surpassed previous state-of-the-art methods across various reduction ratios in both image and video understanding tasks.

Limitation. Although VisionTrim achieves 96.9% of the original performance with an 88.9% reduction ratio in token count without additional training costs, it is not entirely without loss. We are committed to advancing our research to develop more efficient approaches with better performance for visual understanding with MLLMs.

Impact Statement

This paper presents a unified framework for vision token compression that accelerates multimodal large language models (MLLMs) in a training-free manner. By reducing visual token redundancy while preserving crucial visual-textual alignment, we significantly improve computational efficiency, especially for image and video multimodal understanding tasks. This enhancement makes MLLMs more scalable and accessible. While the immediate societal impact may be limited, as these models scale, they could drive advancements in fields such as intelligent agents and autonomous systems, raising concerns about job displacement and ethical issues related to AI decision-making.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023b.

Cha, J., Kang, W., Mun, J., and Roh, B. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, pp. 13817–13827, 2024.

Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., and Chang, B. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, pp. 19–35. Springer, 2024a.

Chen, Z., Wang, W., Tian, H., Ye, S., Gao, Z., Cui, E., Tong, W., Hu, K., Luo, J., Ma, Z., et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.

Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., Li, K., Sun, X., and Ji, R. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, pp. 6904–6913, 2017.

Gurari, D., Li, Q., Stangl, A. J., Guo, A., Lin, C., Grauman, K., Luo, J., and Bigham, J. P. Vizwiz grand challenge: Answering visual questions from blind people. In *CVPR*, pp. 3608–3617, 2018.

Hu, W., Dou, Z.-Y., Li, L. H., Kamath, A., Peng, N., and Chang, K.-W. Matryoshka query transformer for large vision-language models. *arXiv preprint arXiv:2405.19315*, 2024.

Hudson, D. A. and Manning, C. D. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pp. 6700–6709, 2019.

Jang, Y., Song, Y., Yu, Y., Kim, Y., and Kim, G. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, pp. 2758–2766, 2017.

Jin, Y., Li, J., Liu, Y., Gu, T., Wu, K., Jiang, Z., He, M., Zhao, B., Tan, X., Gan, Z., et al. Efficient multimodal large language models: A survey. *arXiv preprint arXiv:2405.10739*, 2024.

Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742. PMLR, 2023a.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pp. 19730–19742. PMLR, 2023b.

Li, W., Yuan, Y., Liu, J., Tang, D., Wang, S., Qin, J., Zhu, J., and Zhang, L. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024b.

Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W. X., and Wen, J.-R. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023c.

Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., and Yuan, L. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *NeurIPS*, 2023.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *CVPR*, pp. 26296–26306, 2024a.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. 2024b.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, 2024c. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, pp. 216–233. Springer, 2025.

Maaz, M., Rasheed, H., Khan, S., and Khan, F. S. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Shang, Y., Cai, M., Xu, B., Lee, Y. J., and Yan, Y. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Vaswani, A. Attention is all you need. *NeurIPS*, 2017.

Vicuna. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://vicuna.lmsys.org/, 2023.

Wang, A., Sun, F., Chen, H., Lin, Z., Han, J., and Ding, G. [cls] token tells everything needed for training-free efficient mllms. *arXiv preprint arXiv:2412.05819*, 2024a.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024b.

Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., and Zhuang, Y. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, pp. 1645–1653, 2017.

Yang, S., Chen, Y., Tian, Z., Wang, C., Li, J., Yu, B., and Jia, J. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024.

Yao, L., Li, L., Ren, S., Wang, L., Liu, Y., Sun, X., and Hou, L. Deco: Decoupling token compression from semantic abstraction in multimodal large language models. *arXiv preprint arXiv:2405.20985*, 2024.

Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., and Wang, L. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., and Tao, D. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, volume 33, pp. 9127–9134, 2019.

Zhang, Q., Cheng, A., Lu, M., Zhuo, Z., Wang, M., Cao, J., Guo, S., She, Q., and Zhang, S. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*, 2024a.

Zhang, Y., Fan, C.-K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024b.

Zhang, Y., Li, B., Liu, h., Lee, Y. j., Gui, L., Fu, D., Feng, J., Liu, Z., and Li, C. Llava-next: A strong zero-shot video understanding model, April 2024c. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

In this part, we provide more details and additional experimental results on our approach. The supplementary material is organized as follows:

- § A: More implementation details;
- § B: Additional experimental results;
- § C: Broader impacts;
- § D: Asset license and consent.

Furthermore, we will fully release the **source code and checkpoints**.

A. More Implementation Details

A.1. Preliminary: Revisiting MLLMs

Inference Progress of MLLM. Existing MLLMs (Liu et al., 2023; Li et al., 2023b; Zhu et al., 2023) are typically composed of three key parts: a visual encoder, a visual projector, and a base LLM. The visual encoder, typically a pre-trained image encoder, converts an input image into a group of distinctive visual embeddings and concatenates an additional `[CLS]` token to align with the textual representation. Then the visual projector translates those visual embeddings into a sequence of visual tokens $\mathbf{X}_{img}$ in LLM's textual embedding space, forming a multimodal instruction by combining with the embeddings of textual instructions $\mathbf{X}_{text}$. Then the LLM generates the response tokens $\mathbf{Y} = \{y_i\}_{i=1}^{L}$ in an auto-regressive manner based on the multi-modal instruction tokens $(\mathbf{X}_{img}, \mathbf{X}_{text})$, which can be formulated as :

$$p(\mathbf{Y}|\mathbf{X}_{img}, \mathbf{X}_{text}) = \prod_{i=1}^{L} p(y_i|\mathbf{X}_{img}, \mathbf{X}_{text}, \mathbf{Y}_{1:i-1}). \tag{15}$$

Computation Budget Estimation. The MLLM decoders typically utilize the causal self-attention mechanism (Vaswani, 2017), where each token can only attend to the past tokens. The self-attention matrix $\boldsymbol{A} \in \mathbb{R}^{L\times L}$, where L is the length of the input sequence, is computed to globally model the dependence relationships between tokens as:

$$\boldsymbol{A} = \text{Attention}(\boldsymbol{Q}, \boldsymbol{K}) = \text{Softmax}\left(\boldsymbol{Q}\cdot\boldsymbol{K}^{\text{T}}/\sqrt{d_k}\right), \tag{16}$$

where $\boldsymbol{Q} \in \mathbb{R}^{L\times D}$ and $\boldsymbol{K} \in \mathbb{R}^{L\times D}$ represent the query and key matrices respectively, and the scalar d_k is the matrix dimension. The computational complexity of each decoder layer includes the computation of the above multi-head attention (MHA) mechanism and feed-forward network (FFN) module. For the whole LLM, the computational indicator FLOPs after the K_{th} layer can be estimated as:

$$\text{FLOPs}_{0:K-1} = K \times (4nd^2 + 2n^2d + 2ndm), \tag{17}$$

where we assume n is the token number, d is the hidden state size and m is the intermediate size of FFN.
The computational cost of MLLM exhibits a quadratic increase with respect to the quantity of input tokens. This underscores the importance of reducing the number of input tokens, particularly visual tokens (normally 20 times more numerous than system and question prompt tokens) while preserving the model's pioneering performance.

A.2. Theoretical Computation Reduction

For computational efficiency analysis, we focus on FLOPs associated with visual tokens in the multimodal LLM, where N denotes the initial number of visual tokens and $K + R$ represents the number of tokens retained after our TGVC module (K tokens for the main visual feature and R tokens for complement). For a single transformer layer with hidden size d and intermediate size m, the theoretical FLOPs reduction ratio F with a token reduction rate $\gamma = (K + R)/N$ is calculated as:

$$F = 1 - \frac{8\gamma Nd^2 + 4(\gamma N)^2d + 6\gamma Ndm}{8Nd^2 + 4N^2d + 6Ndm} \tag{18}$$

Based on the theoretical reduction formula presented above, it is evident that the computational cost of LLMs scales quadratically with the number of input visual tokens. This observation further underscores the importance of pruning and compressing redundant visual tokens during the forward propagation process of MLLMs.

Table 9. Token number settings for VisionTrim in LLaVA-1.5 (Liu et al., 2024b)

	Retain 32 tokens		Retain 64 tokens		Retain 128 tokens		Retain 192 tokens	
	DVTS module	TGVC module	DVTS module	TGVC module	DVTS module	TGVC module	DVTS module	TGVC module
LLaVA-1.5	24	8	48	16	96	32	144	48

Table 10. Token number settings for VisionTrim in LLaVA-NeXT (Liu et al., 2024c)

	Retain 80 tokens		Retain 160 tokens		Retain 320 tokens		Retain 640 tokens	
	DVTS module	TGVC module	DVTS module	TGVC module	DVTS module	TGVC module	DVTS module	TGVC module
LLaVA-NeXT	70	10	140	20	280	40	560	80

B. Additional Experimental Results

B.1. Token Number Settings

For LLaVA-1.5 (Liu et al., 2024b), We report the number of tokens that capture the primary visual information within the DVTS module and the number of tokens utilizing textual information for vision complementarity in the TGVC module across four configurations: 32 tokens, 64 tokens, 128 tokens, and 192 tokens. These values are summarized in Table 9. Additionally, for LLaVA-NeXT (Liu et al., 2024c), which consists of five subfigures, we present the number of tokens retained in the DVTS module and the number of visual complement tokens in the TGVC module across the same four configurations, as shown in Table 10. In the case of Video-LLaVA (Lin et al., 2023), which processes 8 frames per video with 256 tokens per frame, we perform token pruning to reduce the frame token count to 17 tokens per frame, resulting in a total of 136 tokens for the entire video.

B.2. More Quantitative Results

We also present quantitative experimental results under reduced visual token counts to assess the efficacy of our proposed method. As illustrated in Table 11, VisionTrim denotes our approach applied directly during inference, without the need for additional training, while VisionTrim‡ refers to the efficient fine-tuning of the cross-modality attention module in TGVC and the projector, using only 1/10th of the LLaVA-1.5 dataset. Notably, even with just 16 tokens, VisionTrim preserves 90% of the original performance in a training-free manner, demonstrating the robustness and effectiveness of our approach. Furthermore, we explore the extreme case of retaining only a single visual token (resulting in a 99.8% reduction in token count), and surprisingly, VisionTrim still maintains approximately 80% of the original performance without any additional training. When fine-tuned with only 1/10th of the LLaVA-1.5 dataset, VisionTrim achieves 86.5% of the original performance. We also observe that as the number of retained tokens decreases and the compression ratio increases, the performance improvements due to fine-tuning become more pronounced. Notably, despite using only a small subset of the dataset for fine-tuning, VisionTrim exhibits remarkable performance, further highlighting its potential for more flexible deployment and broader applicability in real-world scenarios.

B.3. More Case Studies

Figure 7 illustrates several examples of VisionTrim's application in both image and video understanding tasks. Unlike other methods, which often rely on a larger number of visual tokens, VisionTrim demonstrates notable efficacy in capturing visual details with a minimal token set. For instance, VisionTrim successfully differentiates between distinct characters in an image and accurately identifies the license plate number on a bus. In the context of video understanding, the Video-ChatGPT (Maaz et al., 2023) model utilizes a fixed number of visual token representations across the entire video, which limits its ability to generate comprehensive descriptions that capture temporal dynamics. On the other hand, Video-LLaVA processes eight frames per video, allocating 256 tokens per frame. However, the substantial visual redundancy inherent in this approach often leads to hallucinations. For example, in the case of a soccer game video, Video-LaVA misinterprets the player's footwork and interactions with defenders. In contrast, VisionTrim is able to retain crucial visual information while significantly reducing redundancy. This not only enhances visual comprehension but also optimizes efficiency, thereby making it a more practical and viable solution for efficient multimodal interaction in real-world applications.

Table 11. The performance of our VisionTrim on different configurations of token counts on LLaVA-1.5-7B (Liu et al., 2024b).

Method	GQA	MMB	MME	POPE	SQA	VQA^{V2}	VQA^{Text}	SEED	MMVet	VizWiz	Avg.
Upper Bound, 576 Tokens (**100%**)											
Vanilla (Liu et al., 2024a)	61.9	64.7	1862	85.9	69.5	78.5	58.2	58.6	31.1	50.1	100%
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	
Retain Averaged 32 Tokens (↓ 94.4%)											
VisionTrim	55.3	60.2	1646	77.5	70.1	72.6	54.7	52.5	30.4	52.2	94.0%
	89.3%	93.0%	88.4%	90.2%	100.9%	92.5%	94.0%	89.6%	97.7%	104.2%	
VisionTrim‡	59.8	65.0	1766	84.1	69.2	75.7	55.8	55.5	32.1	52.3	98.4%
	96.6%	100.5%	94.8%	97.9%	99.6%	96.4%	95.9%	94.7%	103.2%	104.4%	↑ (4.4%)
Retain Averaged 16 Tokens (↓ 97.2%)											
VisionTrim	51.5	57.7	1510	72.6	69.4	68.4	53.0	53.4	28.9	50.0	90.0%
	83.2%	89.2%	81.1%	84.5%	99.9%	87.1%	91.1%	91.1%	92.9%	99.8%	
VisionTrim ‡	55.0	61.3	1612	78.1	70.2	72.3	55.4	55.9	30.2	50.5	94.3%
	88.9%	94.7%	86.6%	90.9%	101.0%	92.1%	95.2%	95.4%	97.1%	100.8%	↑ (4.3%)
Retain Averaged 8 Tokens (↓ 98.6%)											
VisionTrim	47.2	50.4	1454	68.4	68.2	62.8	51.5	52.1	27.3	49.5	85.4%
	76.3%	77.9%	78.1%	79.6%	98.1%	80.0%	88.5%	88.9%	87.8%	98.8%	
VisionTrim ‡	51.7	58.3	1525	75.2	69.9	70.2	55.3	54.4	30.4	49.9	91.8%
	83.5%	90.1%	81.9%	87.5%	100.6%	89.4%	95.0%	92.8%	97.7%	99.6%	↑ (6.4%)
Retain Averaged 4 Tokens (↓ 99.3%)											
VisionTrim	43.2	44.3	1422	62.3	67.8	63.5	48.1	52.2	23.8	50.5	81.5%
	69.8%	68.5%	76.4%	72.5%	97.6%	80.9%	82.6%	89.1%	76.5%	100.8%	
VisionTrim ‡	48.9	48.2	1496	65.4	68.6	72.5	52.0	53.6	29.6	51.4	88.0%
	79.0%	74.5%	80.3%	76.1%	98.7%	92.4%	89.3%	91.5%	95.2%	102.6%	↑ (6.5%)
Retain Averaged 1 Tokens (↓ 99.8%)											
VisionTrim	41.6	43.4	1395	61.9	66.7	61.8	47.5	51.7	23.7	48.2	79.8%
	67.2%	67.1%	74.9%	72.1%	96.0%	78.7%	81.6%	88.2%	76.2%	96.2%	
VisionTrim ‡	47.3	47.4	1435	64.6	68.5	70.2	51.3	53.3	29.3	50.7	86.5%
	76.4%	73.3%	77.1%	75.2%	98.6%	89.4%	88.1%	91.0%	94.2%	101.2%	↑ (6.7%)

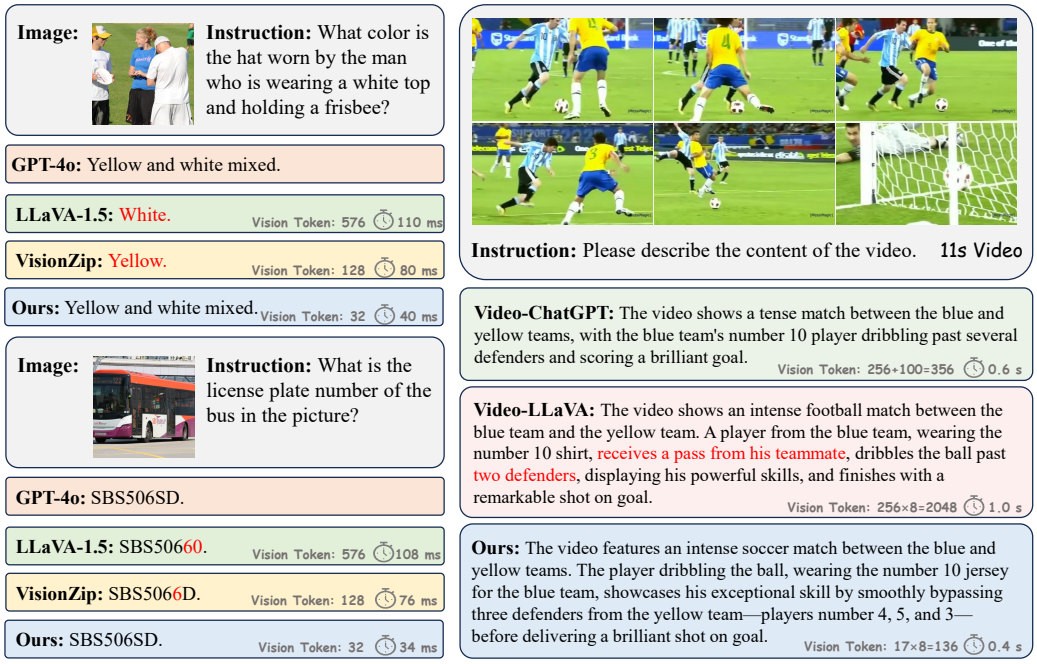
B.4. More Qualitative Results

B.4.1. Visualization of Redundancy in the Vision Encoding Stage

MLLMs typically leverage the encoded feature from the penultimate layer of the vision encoder. To further illustrate the redundancy within vision encoders, we present additional visualization results. As shown in Figure 8, only a small subset of tokens receives high attention, capturing the key visual information, while the majority of visual tokens are allocated minimal attention and contribute little to the overall visual content. This observation underscores the significant redundancy among visual tokens, emphasizing the necessity for their compression.

B.4.2. Visualization of Redundancy in the LLM Decoding Stage

A similar pattern of redundancy is observed during the LLM decoding process. To examine this phenomenon further, we provide visualizations of the attention scores at the 16th and 32nd layers of the LLM as illustrated in Figure 9 and Figure 10. Consistent with the previous findings, only a limited number of tokens receive high attention, while most visual tokens remain largely ignored. This suggests that many of these less relevant visual tokens should be pruned during the LLM



(a) Case of image understanding **(b) Case of video understanding**

Figure 7. An example showcasing VisionTrim's capability to capture visual details in both images and video. The output highlighted in red indicates factual errors.

propagation phase.

B.4.3. Visualization of Attention Distribution Across Both Stages

Based on the observations presented above, a natural question arises: What are the underlying causes of redundancy in visual tokens? In this section, we present a comprehensive analysis of attention changes within both vision encoders and LLMs. As illustrated in Figure 11, attention in the shallow layers is broadly distributed across the entire image. However, as the model progresses to the middle layers, attention rapidly consolidates around a smaller subset of tokens. In the deeper layers, attention becomes increasingly concentrated on a limited number of dominant tokens, reaching its peak concentration by the 23rd layer, which is then utilized for visual token extraction for the LLM. A similar trend is observed during the LLM decoding phase. As shown in Figure 12, attention is initially distributed relatively evenly across all visual tokens. However, as the number of layers increases, the LLM's attention progressively narrows to focus on only a few tokens, neglecting the majority of visual tokens.

Furthermore, to qualitatively assess the effectiveness of our proposed DVTS and TGVC modules, we visualize the attention maps across all 32 layers during the LLM decoding stage, both with and without VisionTrim. We set VisionTrim to reduce the vanilla 576 visual tokens to 128 tokens in the vision encoding stage before being input to the LLM. For comparison, we randomly select 128 visual tokens for the baseline LLM. As shown in Figure 13, the vanilla LLaVA-1.5-7B model exhibits the aforementioned redundancy in visual tokens, particularly after the shallow layers (i.e., the first two layers), where attention is concentrated on only a small subset of tokens, largely disregarding the majority of visual tokens. In contrast, as depicted in Figure 14, VisionTrim improves the cross-modal alignment between visual and textual tokens, enabling the LLM to more effectively focus on the retained image tokens while preserving the most salient visual information. This enhancement not only maintains model performance but also significantly boosts inference speed and efficiency.

C. Broader Impacts

This paper introduces a training-free method, named VisionTrim, designed to accelerate Multimodal Large Language Models (MLLMs) through two plug-and-play modules. These modules effectively filter the essential visual tokens, maintaining both global semantics and local continuity, while leveraging textual information to guide token merging, thereby enhancing the visual-textual alignment. On the positive side, our approach has the potential to significantly benefit the efficient deployment

Table 12. Open-source resources utilized in this paper.

Name	License	URL
TextVQA Dataset	BSD License	https://textvqa.org/
VQA Dataset	BSD License	https://github.com/GT-Vision-Lab/VQA
ScienceQA Dataset	MIT License	https://github.com/lupantech/ScienceQA
POPE Dataset	MIT License	https://github.com/AoiDragon/POPE
LLaVA-1.5	Apache License 2.0	https://github.com/haotian-liu/LLaVA
LLaVA-NeXT	Apache License 2.0	https://github.com/LLaVA-VL/LLaVA-NeXT
Video-LLaVA	Apache License 2.0	https://github.com/PKU-YuanGroup/Video-LLaVA
MMBench Dataset	Apache License 2.0	https://github.com/open-compass/MMBench
MME	Apache License 2.0	https://github.com/BradyFU/Awesome-Multimodal-Large-Language-Models
MM-Vet Dataset	Apache License 2.0	https://github.com/yuweihao/MM-Vet
SEED-Bench Dataset	Apache License 2.0	https://github.com/AoiDragon/POPE
COCO Dataset	Creative Commons Attribution 4.0	https://cocodataset.org/#home
VizWiz Dataset	Creative Commons Attribution 4.0	https://vizwiz.org/tasks-and-datasets/vqa/
GQA Dataset	---	https://cs.stanford.edu/people/dorarad/gqa/index.html
OCR-VQA	---	https://ocr-vqa.github.io/

of MLLMs for real-world image and video understanding tasks, offering a clear reduction in training and inference costs while maintaining competitive performance. However, due to the inherent robustness challenges of large multimodal models, some erroneous outputs may result in misinformation or safety concerns. To mitigate these risks, we recommend implementing a stringent security protocol to address potential failures of our approach in practical multimodal applications.

D. Asset License and Consent

We conduct an extensive evaluation across 10 widely-used image-based benchmarks and 4 widely used video-based multi-modal understanding tasks to assess the multi-modal understanding and reasoning capability of our proposed VisionTrim. Additionally, we utilize 66K mixture instruction following data (a subset of the original 665K LLaVA-1.5 dataset) for efficient instruction tuning. All the datasets are publicly and freely available for academic research. Table 12 provides a list of the resources that have been used in this research paper and their associated licenses.

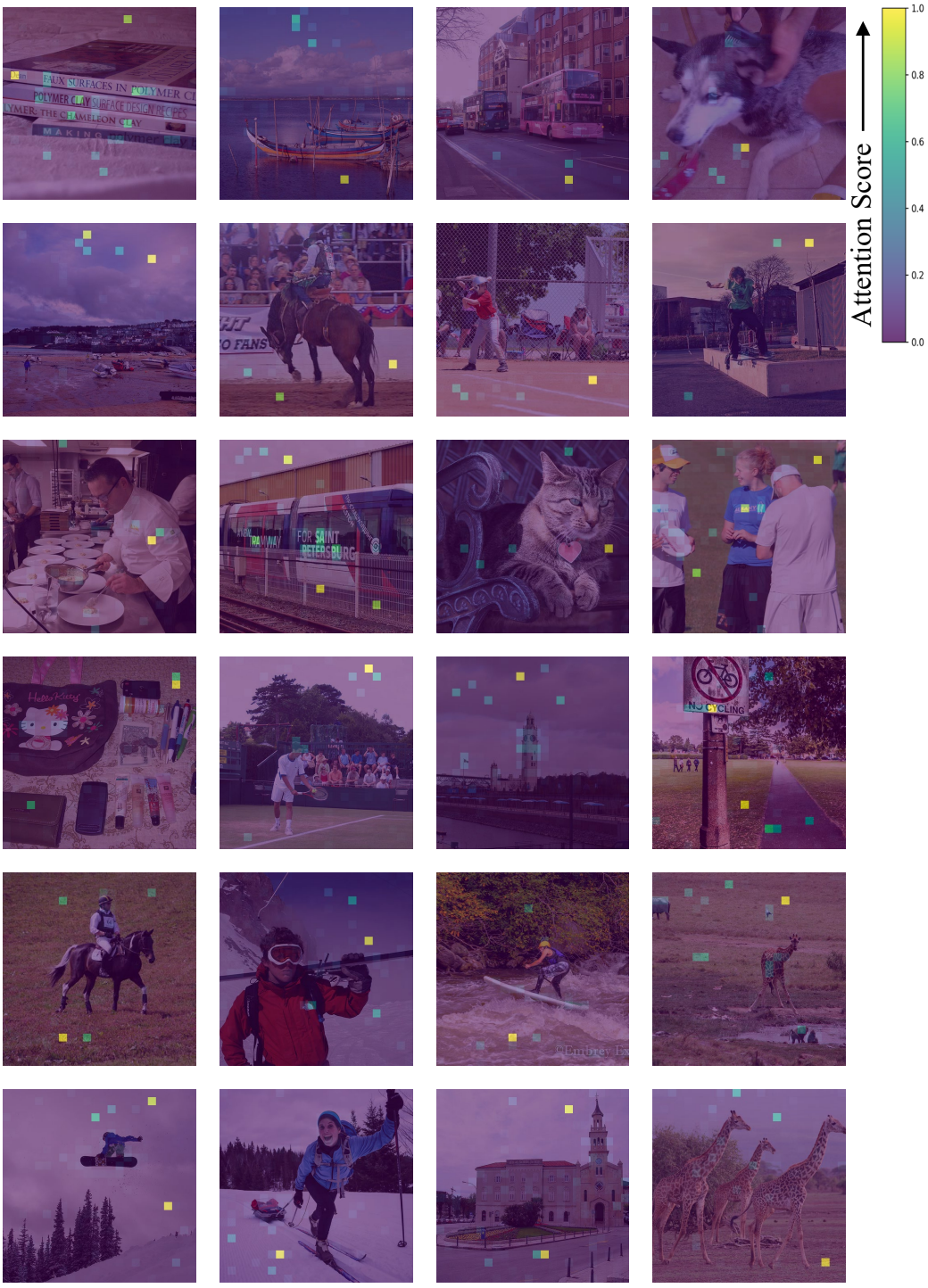


Figure 8. Visualization of redundancy in the penultimate layer of vision encoder (CLIP Model).

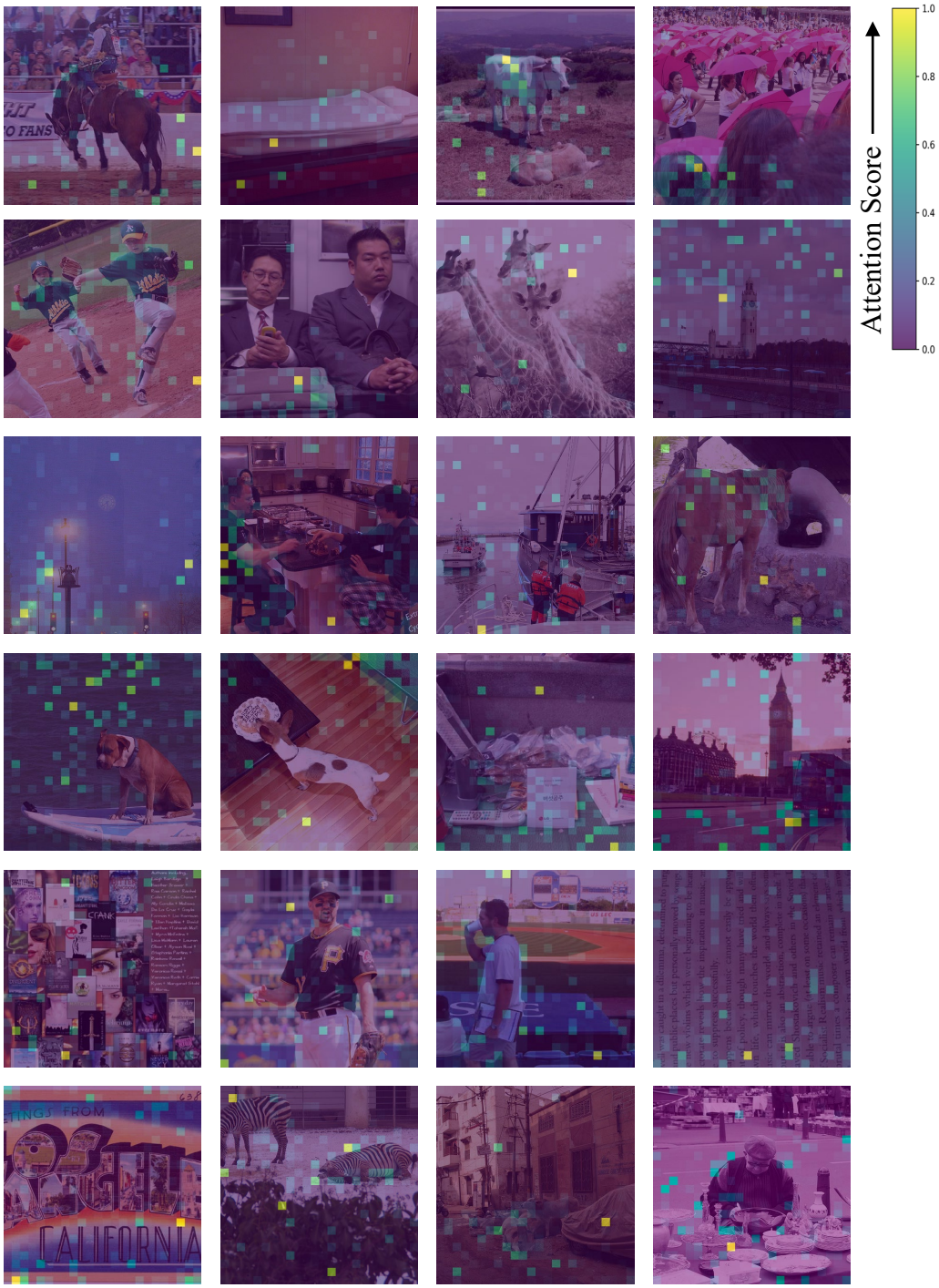


Figure 9. Visualization of redundancy in the 16th layer of LLM.

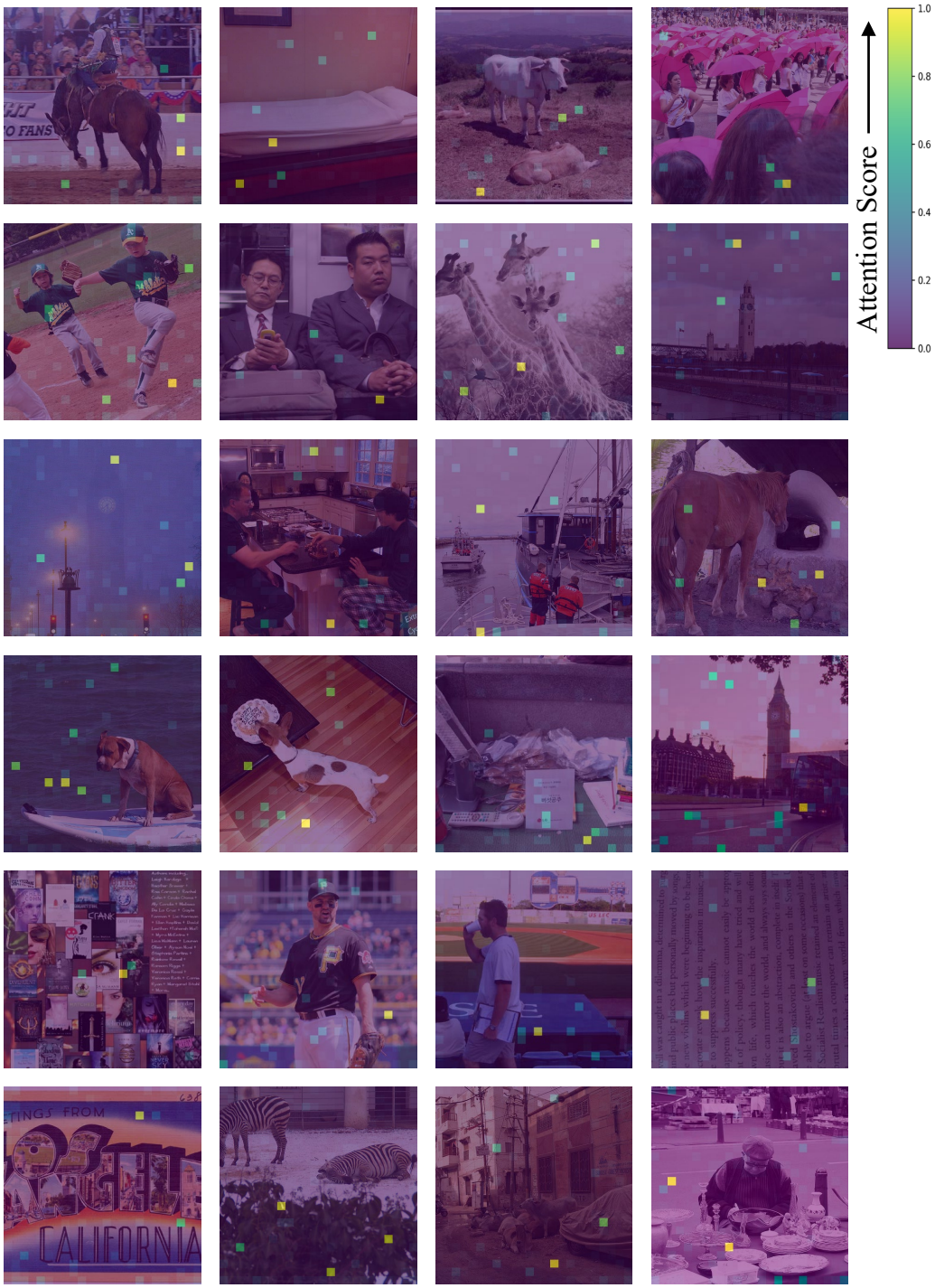


Figure 10. Visualization of redundancy in the 32nd (final) layer of LLM.

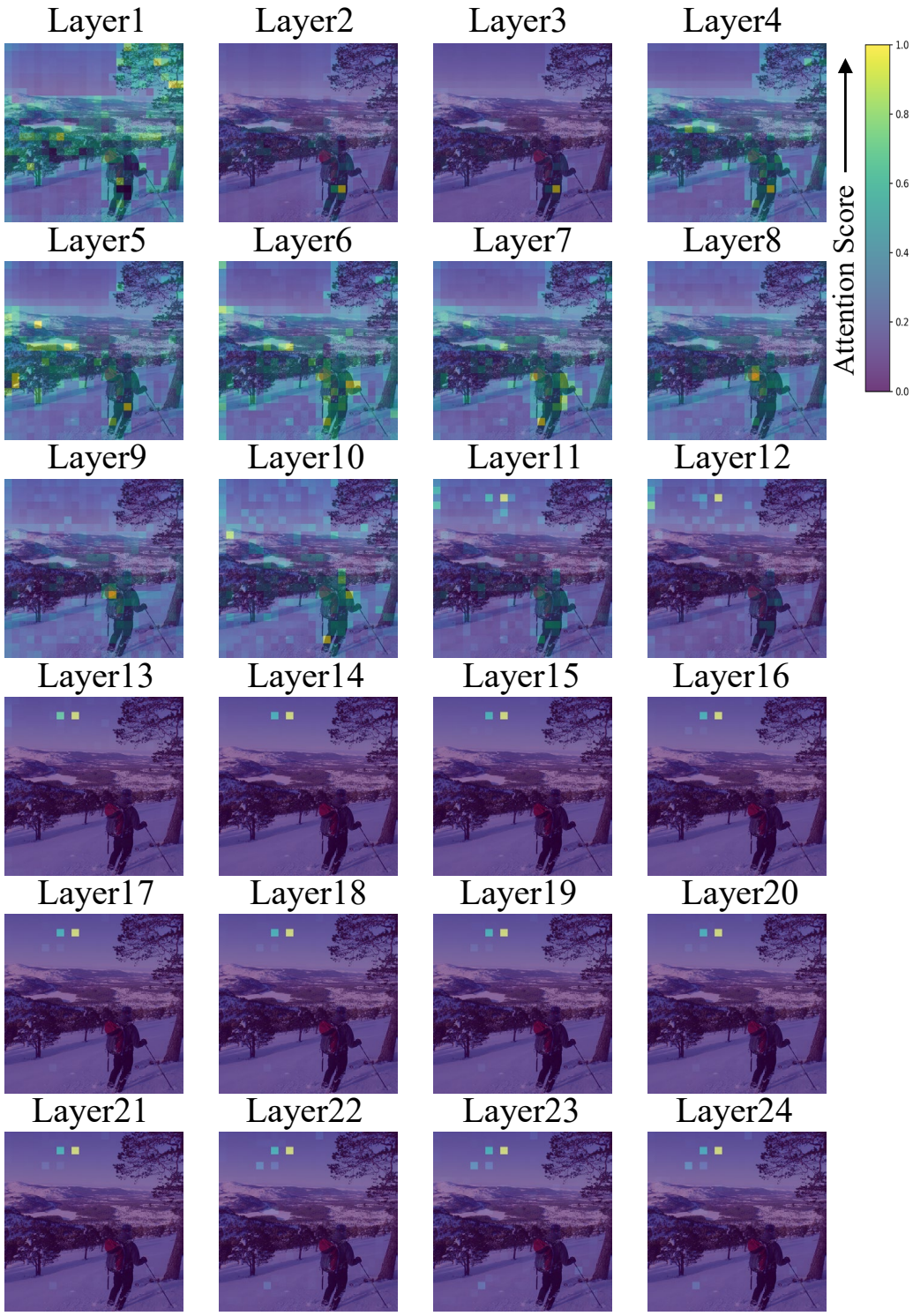


Figure 11. Visualization of attention distribution change in vision encoder (CLIP Model).

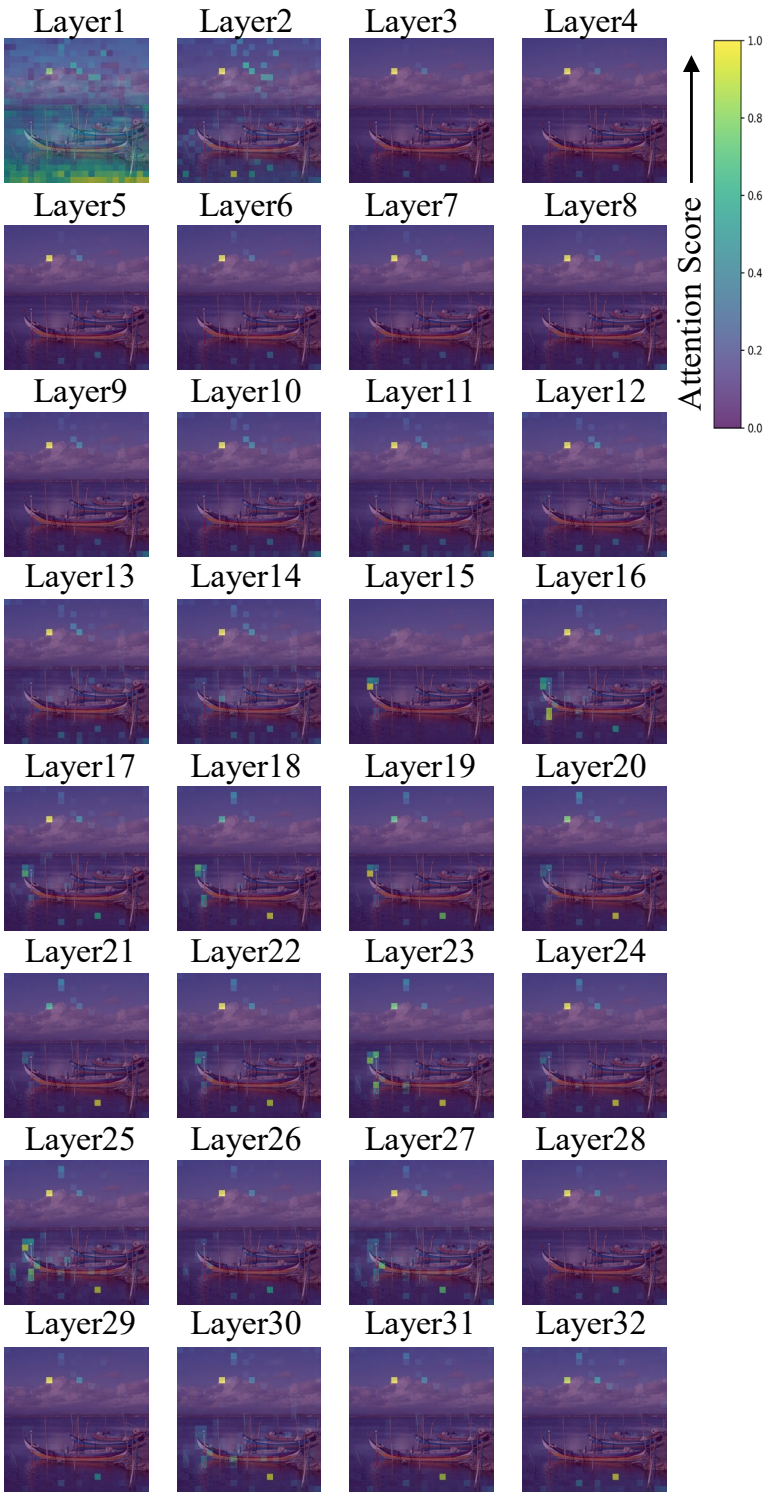


Figure 12. Visualization of attention distribution change in LLM.

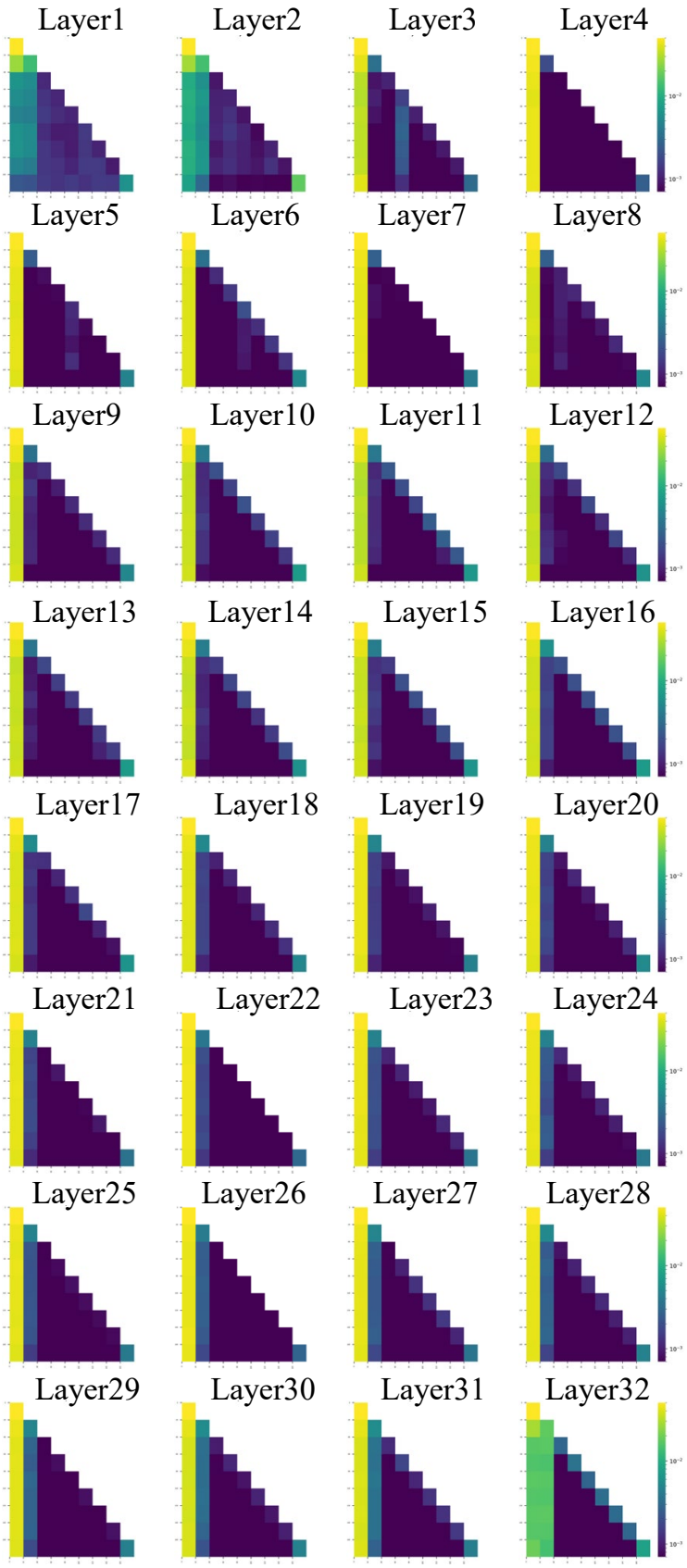


Figure 13. Visualization of attention maps of all 32 layers in Vanilla LLM processing.

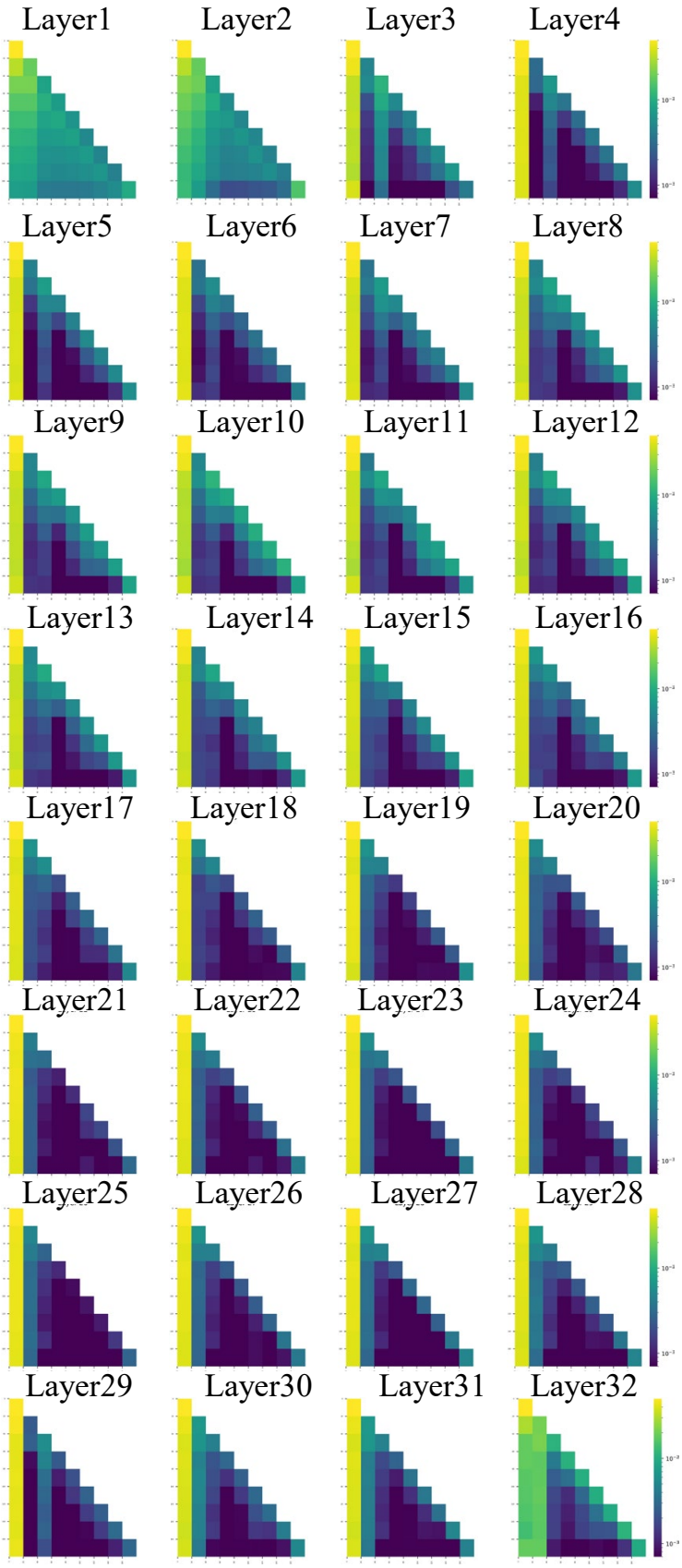


Figure 14. Visualization of attention maps of all 32 layers in LLM processing with our proposed VisionTrim.