SinkPruner: Sink-Free Visual Token Pruning for Multimodal Large Language Models

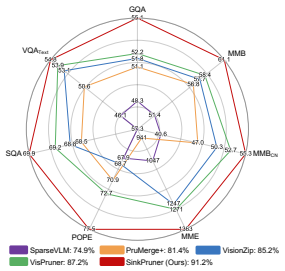
Shiyu Li[*1], **Zi-Yuan Hu**[*1], **Shijia Huang**[2], **Yanyang Li**[2], **Yiwu Zhong**[3], **Liwei Wang**[#1]
[1]The Chinese University of Hong Kong [2]Weitu AI [3]Peking University

Abstract

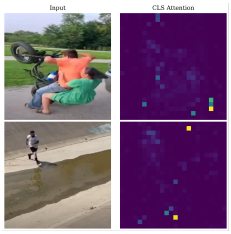
Despite their strong multimodal understanding ability, multimodal large language models (MLLMs) incur substantial computational overhead when processing long visual token sequences. To reduce inference costs, recent studies have explored visual token pruning through vision-centric or text-guided strategies. However, these methods often overlook high-norm outlier tokens, *i.e.*, tokens with abnormally large feature norms, leading to suboptimal pruning decisions. In this work, we show that such high-norm outlier tokens are highly redundant in both feature and spatial dimensions, yet are often mistakenly preserved as informative cues by existing methods. Motivated by this observation, we propose **SinkPruner**, a training-free visual token pruning framework for efficient MLLM inference. *SinkPruner* follows a coarse-to-fine design with two key modules: a **visual sanitizer** that filters high-norm redundancies and alleviates attention sink and attention dispersion, and a **text-guided pruner** that further retains tokens semantically aligned with the text query. Extensive experiments on twelve image-language and four video-language benchmarks demonstrate the effectiveness, efficiency, and generalizability of our framework. Notably, *SinkPruner* preserves 96.5% (91.8%) of the original performance of LLaVA-1.5 (Qwen2.5-VL) under an 89% token reduction. Experiments further indicate that our *visual sanitizer* exhibits promising transferability in enhancing the performance of existing pruning methods. Our code is available at https://github.com/LaVi-Lab/SinkPruner.

1 Introduction

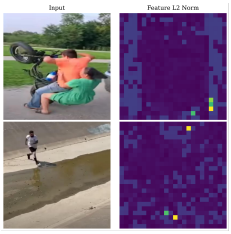
Recent advances in large language models (LLMs) (OpenAI, 2023; Bai et al., 2023a; Touvron et al., 2023) have catalyzed a new generation of multimodal LLMs (MLLMs) (Li et al., 2024c; Bai et al., 2023b; Li et al., 2023a; Liu et al., 2024a), achieving remarkable capabilities in both image and video understanding (Guo et al., 2023; Zhao et al., 2024; Kuang et al., 2025; Jie et al., 2024; Ren et al., 2024). To align visual modalities with textual semantics, MLLMs typically project images or videos into extensive sequences of visual tokens, which are then jointly processed by the LLM decoder (Liu et al., 2024a; Bai et al., 2025b; Touvron et al., 2023). However, this paradigm suffers from a practical bottleneck: the quadratic computational complexity of Transformer attention mechanisms (Vaswani et al., 2017; Dosovitskiy et al., 2021), compounded by excessively long visual sequences, incurs prohibitive computational and memory costs during MLLM inference. This ineffi-



(a) Performance



(b) Attention Outliers



(c) L_2 Norm

Figure 1: **Motivation and Performance Overview.** (a) *SinkPruner* achieves better performance trade-offs than SOTA methods. (b) Attention maps reveal that peaky outliers (bright spots) often appear in background regions, distracting the model. (c) Visual feature L_2 norm heatmap. These attention outliers spatially align with tokens of abnormally high norms (bright spots), confirming that high attention is driven by **high-norm outliers** rather than semantic importance.

[*]equal contributions, [#] corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 13776–13799
October 24-29, 2026 ©2026 Association for Computational Linguistics

ciency restricts the context window availability for textual reasoning and critically hinders deployment in resource-constrained environments, such as edge computing and robotics (Sharshar et al., 2025; Kim et al., 2024; Liu et al., 2024c; Qu et al., 2025).

To alleviate the token burden, prior studies have explored visual token pruning and compression methods for efficient MLLM inference, which can be broadly categorized into vision-centric and text-guided strategies. Specifically, vision-centric methods (*e.g.*, VisionZip (Yang et al., 2025b), HoloV (Zou et al., 2025)) operate within the vision encoder and remove redundancy based on visual cues, such as *CLS attention* or feature similarity among visual tokens. Text-guided methods (*e.g.*, FastV (Chen et al., 2024a), SparseVLM (Zhang et al., 2025b)) leverage the LLM decoder to select tokens that appear semantically aligned with the user query, typically via text-visual attention.

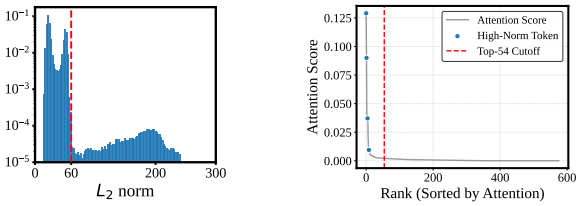
Despite their progress, existing pruning methods suffer from critical limitations rooted in two pathological phenomena within the MLLM pipeline. First, attention-based pruning at the vision encoder is misled by high-norm outliers, *i.e.*, outlier tokens characterized by abnormally large feature norms. State-of-the-art methods heavily rely on *CLS attention* scores (Yang et al., 2025b; Zhang et al., 2025a, 2024) to retain important tokens. However, this criterion systematically prioritizes *high-norm outliers*—tokens that originate from non-informative background regions yet attract abnormally high attention, as shown in Figs. 1b and 1c. Because these high-norm outlier tokens are highly redundant in both feature and spatial dimensions, retaining them severely wastes the limited token budget and suppresses the preservation of truly informative, low-norm tokens. Second, text-guided pruning at the LLM decoder is harmed by attention dispersion and attention sink. When utilizing text-visual attention to select query-relevant visual tokens, the LLM decoder is frequently plagued by *massive activations (attention sinks)* (Kang et al., 2025) and *text-visual attention dispersion* (Zhang et al., 2025a; Zou et al., 2025). Consequently, the decoder disproportionately anchors its attention onto a small set of semantically meaningless sink tokens, while its attention over the remaining tokens becomes highly dispersed. This noise prevents the model from forming a confident ranking of query-relevant regions, rendering attention-based selection highly unreliable, especially at aggressive compression ratios.

To mitigate these limitations, we propose **SinkPruner**, a *training-free, cascading* pruning framework that operates on a coarse-to-fine principle. *SinkPruner* fundamentally addresses the aforementioned bottlenecks by conditioning the visual stream before language decoding. Specifically, it first employs a **visual sanitizer** to preemptively filter high-norm outliers to yield a purified and diverse set of informative tokens. Following this purification, *SinkPruner* applies a **text-guided pruner** at the early layers of the LLM decoder. This pre-filtering effectively reduces attention sink (Kang et al., 2025) and attention dispersion, enabling the cross-modal attention mechanism to reliably identify and retain tokens that are genuinely aligned with the textual instruction.

Comprehensive experiments demonstrate the effectiveness, efficiency, and generalizability of *SinkPruner* across a wide range of image and video understanding tasks. Seamlessly integrated into distinct MLLM architectures—including the fixed-grid LLaVA family and dynamic-resolution Qwen2.5-VL—*SinkPruner* yields superior accuracy-latency trade-offs compared to SOTA methods, as summarized in Fig. 1a. Notably, it enables an 88.9% visual token reduction while preserving 96.5% performance on LLaVA-1.5, unlocking substantial inference speedups for resource-constrained environments. Moreover, our proposed *visual sanitizer* exhibits strong transferability; it functions as an orthogonal, plug-and-play module that can be integrated to consistently enhance existing vision-centric pruning frameworks. Finally, through in-depth empirical analysis, we validate that *SinkPruner* successfully reduces massive activations—cutting the ratio of attention sink tokens within the LLM decoder by about 73%—and significantly lowers the entropy of text-visual attention, thereby empowering the downstream *text-guided pruner* to operate with unprecedented confidence and precision.

2 Related Work

MLLMs and Their Challenges. The rapid evolution of MLLMs (Liu et al., 2023a; Li et al., 2024c, 2023a; Lin et al., 2024b; Zhang et al., 2025c; Ye et al., 2025; Hu et al., 2023; Maaz et al., 2024; Deitke et al., 2025; Wang et al., 2025b; Liu et al., 2025; Dai et al., 2023; Zheng et al., 2025) has revolutionized visual understanding by bridging powerful LLMs (Brown et al., 2020; Touvron et al., 2023;



(a) Feature norm distribution

(b) Attention rank of high-norm tokens

Figure 2: **Characterizing High-Norm Tokens.** (a) The norm histogram shows a clear two-regime separation: most tokens lie below the cutoff, while a small group of high-norm outliers forms a distinct high-value mode. (b) Despite being outliers, high-norm tokens consistently receive the highest attention.

OpenAI, 2023, 2024, 2025; Team, 2023; Reid et al., 2024; Team, 2025; Yang et al., 2024, 2025a) with advanced vision encoders (Radford et al., 2021; Zhai et al., 2023; Fang et al., 2023; Bolya et al., 2025; Liu et al., 2021). Representative architectures (*e.g.*, LLaVA family (Liu et al., 2024a,b; Li et al., 2025) and Qwen-VL series (Bai et al., 2023b, 2025b,a)) have demonstrated enhanced adaptability across diverse multimodal tasks. To mitigate visual hallucinations and improve fine-grained perception (Huang et al., 2024; Jiang et al., 2024), recent trends favor increasing input resolution and supporting dynamic aspect ratios. However, this strategy inevitably exacerbates the token explosion problem. For instance, while LLaVA-1.5 (Liu et al., 2024a) encodes a standard image into 576 tokens, high-resolution models like LLaVA-NeXT (Liu et al., 2024b) scale this to over 2,880 tokens—an order of magnitude longer than typical text prompts. The situation becomes even more prohibitive for video understanding; a 1-hour video sampled at 1fps can exceed 2 million visual tokens (Lin et al., 2024a). These massive visual sequences occupy a disproportionate share of the LLM's context window, highlighting the urgent need for efficient visual token compression strategies.

Visual Token Compression and Pruning. To mitigate the inefficiency caused by excessively long token sequences, extensive research has explored token compression methods across various domains, including natural language processing (Goyal et al., 2020; Wang et al., 2021; Ye et al., 2021; Kim et al., 2022) and computer vision (Rao et al., 2021; Yin et al., 2022; Wei et al., 2023; Liu et al., 2023b; Tang et al., 2023; Dong et al., 2023; Fayyaz et al., 2022; Wang et al., 2024; Hu et al., 2024). In the context of MLLMs, recent studies have primarily focused on reducing visual redundancy (Lin et al., 2025; Xing et al., 2025; Zhong et al., 2024; Tao et al., 2025), which can be broadly categorized into vision-centric and text-guided strategies. Early vision-centric approaches like ToMe (Bolya et al., 2023) and PruMerge (Shang et al., 2025) employ token merging or clustering based on feature similarity to reduce sequence length without training. Building on this, importance-based pruning methods such as VisionZip (Yang et al., 2025b), Vispruner (Zhang et al., 2025a), and FasterVLM (Zhang et al., 2024) prioritize tokens exhibiting high correlations to the [CLS] token. To further optimize the token dropping process, recent methods like MustDrop (Liu et al., 2024d) and PDrop (Xing et al., 2024) introduce progressive dropping mechanisms to adaptively trim the visual sequence, while works like HoloV (Zou et al., 2025) utilize a holistic mechanism to retain global context during aggressive pruning. More recently, HiDrop (Wu et al., 2026) schedules *when* visual tokens enter, are pruned, and exit the LLM layers via late vision injection and early vision exit, while AutoPrune (Wang et al., 2025a) instead decides *how many* tokens to keep per layer by mapping visual-text mutual information to a budget-constrained retention curve. Parallel to vision-centric methods, text-guided approaches leverage the LLM's semantic capabilities to select tokens. FastV (Chen et al., 2024a) and SparseVLM (Zhang et al., 2025b) focus on query-aware token selection using decoder attention scores or cross-modal guidance. Despite their diverse strategies, these methods often overlook the pathological high-norm outlier tokens in vision encoders (Darcet et al., 2024) or the attention sink phenomenon in LLM decoders. In contrast, our *SinkPruner* adopts a cascading framework that preemptively filters inherent visual redundancy and high-norm outliers to clarify the attention landscape, thereby enabling more precise and confident text-guided pruning.

3 Methodology

In this section, we detail the motivation and architecture of **SinkPruner**, a novel, training-free visual token pruning framework for efficient MLLM inference. First, we analyze efficiency bottlenecks of MLLMs, highlighting the critical need to compress extensive visual token sequences. We then present key empirical observations revealing that **high-norm outlier tokens** are inherently redundant and act as attention sinks that disrupt vision-centric

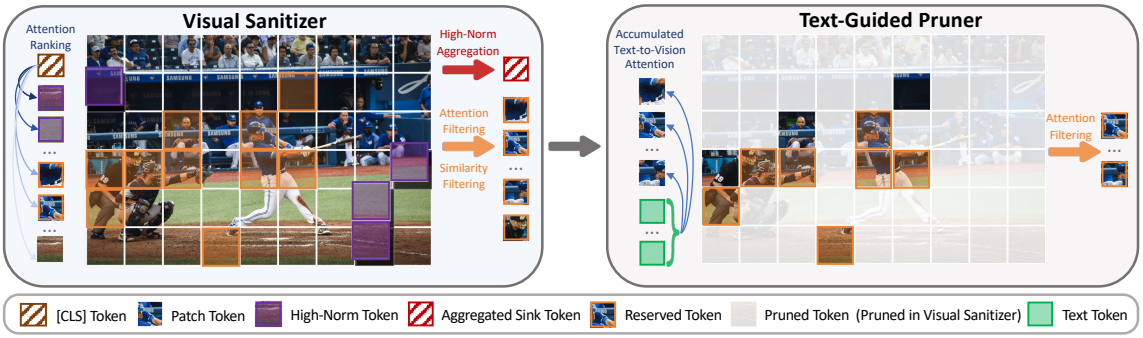
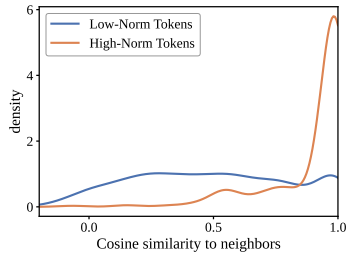


Figure 3: **Overview of *SinkPruner* framework**. **Left (Visual Sanitizer):** As shown in the *Attention Ranking* (far left), *high-norm outlier tokens* (purple) dominate the top ranks despite being background noise. We aggregate these outliers into a single *sink token* (red hashed) and select informative *reserved tokens* (orange) via attention and similarity filtering. **Right (Text-Guided Pruner):** The purified visual sequence further interacts with *text tokens* (green), utilizing accumulated text-to-vision attention to retain only semantically relevant visual tokens.

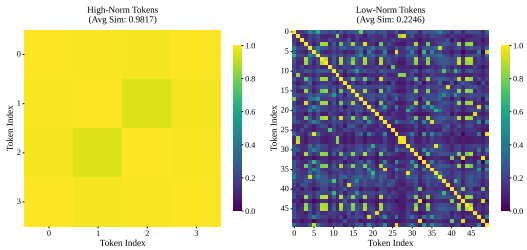
pruning. Motivated by these insights, we introduce our coarse-to-fine cascading *SinkPruner*, as shown in Fig. 3, which features a **visual sanitizer** and a **text-guided pruner**. Specifically, the *visual sanitizer* first preemptively filters high-norm outliers to yield a purified and diverse set of informative tokens, and the *text-guided pruner* subsequently retains visual tokens that are semantically aligned with the text query.

3.1 Preliminary

Architecture of MLLMs. Existing MLLMs generally consist of a vision encoder and an LLM, both built on the Transformer architecture (Vaswani et al., 2017). To be specific, the vision encoder (*e.g.*, CLIP-ViT (Radford et al., 2021)) applies self-attention over visual tokens $\boldsymbol{X}_{vis} = [\boldsymbol{x}_{\text{CLS}}; \boldsymbol{x}^{1}_{\text{img}}, \ldots, \boldsymbol{x}^{n_{vis}-1}_{\text{img}}]$, where n_{vis} denotes the total number of visual tokens. The attention matrix is computed as $\boldsymbol{A}_{vis} = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d})$, where $\boldsymbol{Q}$ and $\boldsymbol{K}$ are the query and key representations, and d represents the hidden dimension. We denote the first row of $\boldsymbol{A}_{vis}$—representing the attention from the [CLS] token to all image patches—as CLS *attention*. Conversely, the LLM processes the concatenated input sequence $\boldsymbol{X} = [\boldsymbol{x}_{\text{sys}}; \boldsymbol{X}_{vis}; \boldsymbol{x}_{\text{txt}}]$ using causal self-attention. Here, $\boldsymbol{x}_{\text{sys}}$ and $\boldsymbol{x}_{\text{txt}}$ denote the system prompt and text instruction with sequence lengths n_{sys} and n_{text}, respectively. This brings the total sequence length to $n = n_{sys} + n_{vis} + n_{text}$. This causal mechanism incorporates rotary position embeddings (RoPE) $\boldsymbol{R}_\theta$ parameterized by θ, along with a lower-triangular causal mask $\boldsymbol{M}$ to ensure unidirectional routing. Accordingly, the LLM attention matrix is computed as $\boldsymbol{A}_{llm} = \text{softmax}\big((\boldsymbol{R}_\theta\boldsymbol{Q})(\boldsymbol{R}_\theta\boldsymbol{K})^T/\sqrt{d} + \boldsymbol{M}\big)$,



(a) Neighbor similarity



(b) Pairwise similarity within each subset

Figure 4: **Feature Redundancy Analysis.** (a) high-norm tokens show distinctively high neighbor similarity. (b) high-norm tokens exhibit extremely high inter-token similarity (feature space collapse), while low-norm tokens maintain high diversity.

and the final output is obtained via $\boldsymbol{O} = \boldsymbol{A}_{llm}\boldsymbol{V}$.

Efficiency Bottleneck in MLLMs. Computing the aforementioned attention matrix $\boldsymbol{A}_{llm}$ and processing the concatenated sequence $\boldsymbol{X}$ through deep LLM layers introduces severe computational overhead. Specifically, for an LLM with T layers and a feed-forward network intermediate width m, the total floating-point operations (FLOPs) can be approximated as $T \times (4nd^2 + 2n^2d + 2ndm)$. This formulation highlights a quadratic computational complexity with respect to the total sequence length n, primarily stemming from the causal self-attention mechanism (*i.e.*, the $2n^2d$ term). Given that $n = n_{sys} + n_{vis} + n_{text}$, this sequence length

in typical MLLM applications is dominated by the visual tokens n_{vis}, which often exceed text prompts by an order of magnitude. Because this massive volume of visual tokens acts as the primary catalyst for the quadratic computational explosion, **reducing the visual token count** n_{vis} emerges as the most essential paradigm for accelerating MLLM inference.

3.2 Redundancy in High-Norm Tokens

As established in Sec. 3.1, reducing visual tokens is essential for efficient MLLM inference. To achieve this, existing pruning methods predominantly rely on attention scores (*e.g.*, CLS *attention*) to gauge token importance, implicitly assuming that tokens with higher attention weights carry richer semantic information. However, the reliability of this assumption remains questionable. Upon inspecting the attention maps within the vision encoder of MLLMs, we observe a counterintuitive phenomenon: clear outliers emerge, where a few non-semantic background regions exhibit unusually peaky attention values. Since attention-based pruning methods are highly sensitive to these peaky values, understanding the true nature of these outliers—and whether they actually encode indispensable visual concepts—is crucial for designing a robust compression strategy.

Outliers are high-norm tokens. We first seek a quantitative way to localize these outliers. By inspecting the vision encoder features, we observe that artifact regions are strongly associated with unusually large feature magnitudes, an effect first reported for self-supervised and supervised ViTs by Darcet et al. (2024), who mitigate it at training time by appending dedicated register tokens. We instead exploit it at inference time as a training-free signal for token pruning. As shown in Figs. 1b and 1c, tokens from artifact patches exhibit much larger ℓ_2 norms than tokens from semantic regions. To further verify this observation, we plot the distribution of token norms over a subset of images in Fig. 2a. The histogram shows a clear two-regime pattern: the majority of tokens concentrate below 60, while a separated group of outliers spans a much higher range (roughly $[60, 250]$), which accounts for only about 1% of all visual tokens. Since the absolute norm scale is encoder-dependent, we identify outliers by their relative rank rather than by a hand-tuned cutoff: given the N visual tokens of an image, we rank them by feature norm and treat the top ρ fraction as **high-norm** tokens, *i.e.*, $\mathbf{X}_{high} = \mathrm{Top}_{\rho}(\{\|x_i\|_2\}_{i=1}^{N})$ with $|\mathbf{X}_{high}| = \lceil \rho N \rceil$, and the remaining tokens $\mathbf{X}_{low} = \mathbf{X} \setminus \mathbf{X}_{high}$ as low-norm tokens. Following the observed ratio above, we set $\rho = 1\%$ throughout, which keeps the criterion applicable to encoders whose norms live on very different scales (see Appendix A). Importantly, these high-norm tokens largely overlap with the observed outliers.

High-norm outlier tokens are spatially redundant. Next, we examine whether high-norm outlier tokens contain informative visual content. We measure the cosine similarity between each token and its 4-nearest spatial neighbors after the patch embedding layer. As shown in Fig. 4a, high-norm outliers typically arise from patches that are highly similar to their neighboring patches. This observation aligns with the visual pattern that these outliers predominantly appear in homogeneous background regions (*e.g.*, sky and walls); see Appendix I for visualization results. Therefore, these tokens encode spatially redundant information and can be discarded without degrading image representation.

High-norm outlier tokens are feature-redundant. We further study redundancy in feature space. Standard attention-based pruning keeps the top-k tokens ranked by CLS *attention*, but this set is heavily dominated by high-norm tokens. We therefore partition the selected tokens into the high-norm subset $\mathbf{X}_{high}$ and the low-norm subset $\mathbf{X}_{low}$ defined above. We then compute pairwise cosine similarity within each subset (Fig. 4b). High-norm tokens show extremely high intra-set similarity, indicating representational collapse. In contrast, low-norm tokens remain more diverse and typically carry richer semantic details; see the Appendix I for visualizations. This confirms that retaining high-norm tokens wastes the token budget on redundant information.

The "attention sink" trap. Although high-norm outlier tokens are largely redundant, they can still dominate self-attention. As shown in Fig. 2b, they consistently receive the highest attention from the [CLS] token, acting as *attention sinks* that absorb a disproportionate amount of global attention. This creates a failure mode for vision-centric pruning methods such as VisionZip (Yang et al., 2025b) and FasterVLM (Zhang et al., 2024): when token importance is determined solely by attention scores, these methods prioritize high-norm outlier tokens while potentially discarding more informative low-

norm tokens.

3.3 SinkPruner Framework

Building upon observations in Sec. 3.2, we propose **SinkPruner**, a training-free, cascading visual token pruning framework designed for efficient MLLM inference, as shown in Fig. 3. Our approach operates on a coarse-to-fine principle: we first introduce a **visual sanitizer** to filter high-norm outliers, yielding a purified and diverse set of informative tokens, and then a **text-guided pruner** to retain visual tokens that are semantically aligned with the text query.

Visual sanitizer. To address the "high-norm trap" where high-norm outlier tokens act as attention sinks, we explicitly isolate these tokens based on feature magnitude. Let $\mathbf{X} = \{x_1, \ldots, x_N\} \in \mathbb{R}^{N \times D}$ denote a sequence of visual tokens extracted from the vision encoder. We first compute the L_2-norm $n_i = \|x_i\|_2$ for each visual token, and apply the scale-free top-ρ rule of Sec. 3.2 to partition the visual sequence into outliers $\mathbf{X}_{high} = \mathrm{Top}_\rho(\{n_i\}_{i=1}^N)$, *i.e.*, the $\lceil \rho N \rceil$ tokens with the largest norms, and candidate tokens $\mathbf{X}_{low} = \mathbf{X} \setminus \mathbf{X}_{high}$, with $\rho = 1\%$ by default. Since this rule ranks tokens against each other instead of against an absolute value, it requires no per-model calibration and transfers directly to backbones whose feature norms live on different scales.

Subsequently, we aggregate the identified high-norm outlier tokens $\mathbf{X}_{high}$ into a single proxy token to compress visual redundancy, while preserving global information. To be specific, as discussed in Sec. 3.2, these tokens are compressed without significant information loss via average pooling due to their extreme feature similarity:

$$x_{sink} = \frac{1}{|\mathbf{X}_{high}|} \sum_{x \in \mathbf{X}_{high}} x. \tag{1}$$

We then employ a hybrid selection strategy to extract a representative subset from the informative candidates $\mathbf{X}_{low}$, balancing semantic salience with spatial coverage. First, to preserve prominent regions, we select the set $\mathbf{X}_{res}$ containing the top-k_{res} tokens with the highest *CLS attention* scores $\mathcal{A}_{cls}$[1]:

$$\mathbf{X}_{res} = \{x \in \mathbf{X}_{low} \mid \mathrm{Rank}(\mathcal{A}_{cls}(x)) \leq k_{res}\}. \tag{2}$$

[1]For encoders without a [CLS] token (*e.g.*, the vision tower of Qwen2.5-VL (Bai et al., 2025b)), $\mathcal{A}_{cls}$ is replaced by the average received visual self-attention, while all other components remain unchanged; see Appendix D.3.

Methods	GQA	MMB	MMB_{CN}	MME	POPE	SQA	VQA_{Text}	Avg
Upper Bound, 576 Tokens	61.9	64.7	58.1	1862	85.9	69.5	58.2	100.0%
LLaVA-1.5-7B	*Retain 64 Tokens* (Pruning Ratio = **88.9%**)							
ToMe (ICLR23)	48.6	43.7	-	1138	52.5	50.0	45.3	59.7
FastV (ECCV24)	46.1	48.0	52.7	1256	48.0	51.1	47.8	74.1
MustDrop (2024.11)	53.1	60.0	53.1	1612	68.0	63.4	54.2	88.6
PDrop (2024.10)	41.9	33.3	50.5	1092	55.9	68.6	45.9	72.5
VisionZip (CVPR25)	55.1	60.1	55.4	1690	77.0	69.0	**55.5**	93.2
SparseVLM (ICML25)	52.7	56.2	46.1	1505	75.1	62.2	51.8	85.4
HoloV (NeurIPS25)	55.3	**63.3**	55.1	1715	80.3	69.5	55.4	94.7
ApET (2026.02)	56.9	61.2	54.4	1714	**84.4**	68.9	53.9	94.6
SinkPruner (Ours)	**57.4**	62.8	**56.9**	**1754**	83.8	**70.0**	**55.5**	**96.5**
LLaVA-1.5-7B	*Retain 32 Tokens* (Pruning Ratio = **94.4%**)							
ToMe (ICLR23)	43.6	31.6	28.1	828	39.0	41.4	38.3	54.7
FastV (ECCV24)	41.5	37.8	33.2	885	32.5	42.6	42.5	57.5
SparseVLM (ICML25)	48.3	51.4	40.6	1047	67.9	57.3	46.1	74.9
PruMerge+ (2024.05)	51.1	56.8	47.0	941	70.9	68.5	50.6	81.4
VisionZip (CVPR25)	51.8	57.7	50.3	1247	68.7	68.8	53.1	85.2
VisPruner (ICCV25)	52.2	58.4	52.7	1271	72.7	69.2	53.9	87.2
SinkPruner (Ours)	**55.1**	**61.2**	**55.3**	**1363**	**78.5**	**69.9**	**54.8**	**91.2**

Table 1: Performance comparison on LLaVA-1.5-7B across different image-language benchmarks under different pruning ratios. Best results are in **bold**. More comprehensive results are provided in the Appendix E.

Next, to capture subtle background semantics and avoid feature collapse, we perform a similarity-based de-duplication on the remaining tokens $\mathcal{R} = \mathbf{X}_{low} \setminus \mathbf{X}_{res}$. We initialize the selected set $\mathcal{S} = \mathbf{X}_{res}$ and iteratively[2] sample a token x^* that exhibits the lowest similarity to the current set $\mathcal{S}$, appending it to $\mathbf{X}_{div}$:

$$x^* = \underset{x \in \mathcal{R}}{\mathrm{argmin}} \left(\max_{s \in \mathcal{S}} \mathrm{CosSim}(x, s) \right). \tag{3}$$

The final purified visual representation $\mathbf{Z}$ is formed by concatenating the aggregated sink token and the selected subsets: $\mathbf{Z} = [x_{sink}, \mathbf{X}_{res}, \mathbf{X}_{div}]$, which is subsequently projected and fed into the LLM.

Text-guided pruner. Leveraging the purified sequence $\mathbf{Z}$, the model effectively evaluates semantic relevance without interference from high-norm outliers. We utilize the text-to-vision attention in an early LLM decoder layer as the importance estimator. Letting L_t denote the number of text tokens, we compute a global relevance score $\tilde{p}_j$ for each visual token $z_j \in \mathbf{Z}$ by aggregating its attention weights across all textual queries:

$$\tilde{p}_j = \frac{1}{L_t} \sum_{i=1}^{L_t} \mathrm{Softmax}(\mathbf{Q}_{text} \cdot \mathbf{K}_{vis}^\top)_{i,j}. \tag{4}$$

We then retain the top-K visual tokens with the highest $\tilde{p}$ scores. This query-aware selection ensures the LLM focuses its computational budget strictly on regions required for reasoning.

4 Experiments

4.1 Experiment settings

Evaluation benchmarks. To comprehensively demonstrate the effectiveness and generalizability

[2]In practice we use a batched approximation of this serial rule; see Appendix D.4.

Methods	MMB	MME	POPE	SQA	VQA_{Text}	Avg
Upper Bound	84.2	2292	86.1	88.78	83.04	100%
Qwen2.5-VL-7B	*Pruning Ratio* = ↓ 66.7%					
FastV (ECCV24)	75.7	2072	82.2	78.5	77.9	91.6
HoloV (NeurIPS25)	78.3	2093	85.0	79.8	78.9	93.6
VisionZip (CVPR25)	75.8	2098	84.4	81.7	70.9	91.4
SinkPruner (Ours)	**82.1**	**2268**	**85.2**	**88.7**	**81.1**	**98.6**
Qwen2.5-VL-7B	*Pruning Ratio* = ↓ 77.8%					
FastV (ECCV24)	74.9	2036	80.7	78.0	69.0	88.5
HoloV (NeurIPS25)	76.5	2043	82.3	79.8	70.3	90.0
VisionZip (CVPR25)	**80.33**	2174	**83.38**	84.23	70.43	93.4
DivPrune (CVPR25)	76.98	2163	80.59	80.91	65.86	90.0
MMTok (2025.08)	79.30	**2217**	82.38	81.61	70.49	92.7
SinkPruner (Ours)	79.7	2207	82.8	**87.7**	**79.2**	**96.3**
Qwen2.5-VL-7B	*Pruning Ratio* = ↓ 88.9%					
FastV (ECCV24)	69.2	1940	78.6	77.4	60.3	83.6
HoloV (NeurIPS25)	72.4	2006	**80.7**	79.5	61.8	86.2
VisionZip (CVPR25)	75.60	2003	78.90	82.30	63.78	87.7
DivPrune (CVPR25)	72.85	1957	74.99	79.57	59.59	84.1
MMTok (2025.08)	74.74	2051	78.75	80.47	63.90	87.5
SinkPruner (Ours)	**77.6**	**2123**	78.3	**87.1**	**70.8**	**91.8**

Table 2: Performance comparison on Qwen2.5-VL-7B with various token compression methods under different pruning ratios. Best results are in **bold**. More comprehensive results are provided in the Appendix.

Method	MVBench Acc	SEEDBench Acc	VideoMME Score	Avg. (%)
Qwen2.5-VL-7B (Full)	68.10	62.18	60.67	100.0%
DART (EMNLP25)	65.80	61.00	57.74	96.6%
DivPrune (CVPR25)	65.85	59.79	57.78	96.0%
SinkPruner (Ours)	**66.70**	**61.80**	**58.59**	**98.0%**

Table 3: Performance comparison of various methods across different video-language benchmarks under an 80% token pruning ratio. More comprehensive experimental results are provided in the Appendix E.

of our *SinkPruner*, we conduct experiments across a wide range of multimodal understanding tasks. For image-language understanding, we evaluate on twelve widely used benchmarks: GQA (Hudson and Manning, 2019), MMBench (MMB) and its Chinese counterpart MMB-CN (Liu et al., 2024e), MME (Fu et al., 2023), POPE (Li et al., 2023b), ScienceQA (SQA) (Lu et al., 2022), VQA-v2 (Goyal et al., 2017), and TextVQA (Singh et al., 2019), which mainly probe perception and short-form reasoning, together with four substantially harder benchmarks that target high-level reasoning rather than perception alone, namely MMStar (Chen et al., 2024b), MMMU (Yue et al., 2024), AI2D (Kembhavi et al., 2016), and MM-Vet (Yu et al., 2024). Furthermore, since video processing inherently involves massive spatiotemporal redundancy, making efficient pruning critical, we also extend our evaluation to video-language understanding. Specifically, we conduct experiments on four representative video benchmarks: MVBench (Li et al., 2024b), SEED-Bench (Li et al., 2024a), NextQA (Xiao et al., 2021), and VideoMME (Fu et al., 2025). We describe each benchmark and the corresponding evaluation metric in Appendix C.1.

Methods	MMStar	MMMU	AI2D	MM-Vet	Avg
Upper Bound	34.04	36.56	55.15	33.99	100%
LLaVA-1.5-7B	*Retain 32 Tokens* (Pruning Ratio = 94.4%)				
FastV (ECCV24)	30.41	35.11	50.49	22.66	85.9
VisionZip (CVPR25)	30.36	**35.67**	51.81	25.23	88.7
SinkPruner (Ours)	**32.14**	35.44	**53.63**	**29.36**	**93.7**

Table 4: Performance comparison on LLaVA-1.5-7B across harder reasoning-oriented benchmarks under an aggressive 32-token budget. Best results are in **bold**.

Implementation Details. We identify high-norm outliers with the scale-free top-ρ rule of Sec. 3.2 and keep $\rho = 1\%$ for every backbone and every benchmark, so no per-model threshold calibration is involved. For low-norm selection, we configure both the salience and diversity pool sizes to match the target token budget (e.g., 64 for *Retain-64* and 32 for *Retain-32*). More experimental settings and baseline details are provided in the Appendix D.

4.2 Main Results

SinkPruner achieves superior effectiveness under aggressive pruning As shown in Tab. 1, we evaluate our proposed *SinkPruner* on LLaVA-1.5-7B under two extremely challenging pruning configurations, retaining only 64 (↓ 88.9%) and 32 (↓ 94.4%) visual tokens. Following prior work (Bolya et al., 2023; Chen et al., 2024a; Liu et al., 2024d; Xing et al., 2024; Zhang et al., 2025b; Yang et al., 2025b; Zhang et al., 2025a; Zou et al., 2025), we report all results in normalized percentage form, where the vanilla 576-token model is treated as the 100% upper bound. In the 64-token setting, *SinkPruner* attains an average performance of **96.5%**, reducing nearly 90% of visual tokens while incurring only a 3.5% drop from the full model. It surpasses strong recent methods: exceeding VisionZip by +3.3% and HoloV by +1.8%. When retaining only 32 visual tokens, *SinkPruner* continues to deliver strong performance with an average of **91.2%**, outperforming all existing pruning methods by a clear margin. Compared with the best prior method VisPruner, *SinkPruner* achieves a +4.0% improvement, and exceeds VisionZip by +6.0%. Even in this extremely low-token regime, our method consistently achieves the top performance across all benchmarks, showing that *SinkPruner* maintains high fidelity even when over 94% of the visual tokens are removed.

SinkPruner generalizes seamlessly to dynamic-resolution architectures. To further demonstrate the generality of *SinkPruner*, we evaluate its performance on Qwen2.5-VL-7B (Bai et al., 2025b),

a state-of-the-art MLLM whose visual pipeline differs fundamentally from the LLaVA family. Unlike LLaVA models that rely on a fixed-grid ViT encoder (*e.g.*, 336×336), Qwen2.5-VL-7B employs a *Naive Dynamic Resolution* mechanism. This architecture allows the model to process images of arbitrary aspect ratios by mapping variable pixel counts to a dynamic number of visual tokens. While this flexibility enhances fine-grained perception for high-resolution inputs, it often results in massive token sequences, leading to significant computational overhead and latent redundancy. Pruning tokens in such a variable-length regime is more challenging, as the model lacks a stable spatial prior. Because this vision tower has no [CLS] token, we instantiate the CLS-free salience score introduced in Sec. 3, which makes *SinkPruner* directly applicable without architectural changes. Our results show that even within this dynamic resolution framework, *SinkPruner* consistently achieves the best average performance, proving its robustness across diverse MLLM architectures. Following the experimental settings in HoloV (Zou et al., 2025), we assess performance under aggressive pruning ratios of 66.7%, 77.8%, and 88.9%. As detailed in Tab. 2, *SinkPruner* demonstrates remarkable robustness, retaining **98.6%**, **96.3%**, and **91.8%** of the full-model performance, respectively. In contrast, existing SOTA methods (*e.g.*, FastV (Chen et al., 2024a), HoloV (Zou et al., 2025), VisionZip (Yang et al., 2025b), DivPrune (Alvar et al., 2025), MM-Tok (Dong et al., 2025)) degrade sharply in this dynamic regime, confirming *SinkPruner*'s superiority in identifying redundancy even within complex, variable-length visual sequences.

SinkPruner preserves performance on harder reasoning tasks. The benchmarks above are largely perception-oriented, so a natural concern is whether pruning preserves perception while destroying the evidence needed for multi-step reasoning. We therefore evaluate on the four harder benchmarks of Sec. 4. As shown in Tab. 4, under the most aggressive 32-token budget *SinkPruner* retains **93.7%** of the full-model average, versus 88.7% for VisionZip and 85.9% for FastV. The gap is largest on MM-Vet (29.36 vs. 25.23 and 22.66) and AI2D (53.63 vs. 51.81 and 50.49), *i.e.*, precisely the tasks requiring several spatially distinct pieces of evidence, while on MMMU all three stay within 0.6 points since that benchmark is dominated by textual knowledge. Sanitizing high-norm tokens therefore does not trade reasoning fidelity for perception fidelity. Results on LLaVA-1.5-13B and the high-resolution LLaVA-NeXT-7B, which verify transfer across model scale and visual-token regimes, are in Appendix F.

Methods	Time (mm:ss)	Prefill (ms)	Latency (ms)	Acc.
Upper Bound (576 Tokens)	19:28	62.75	121.0	100%
LLaVA-1.5-7B	*Pruning Ratio* = ↓ 90.0%			
FastV (ECCV24)	12:02	32.2	75.0	65.5%
VisionZip (CVPR25)	12:24	36.9	77.4	91.8%
SparseVLM (ICML25)	12:20	38.6	77.0	90.5%
SinkPruner (Ours)	12:59	37.1	86.0	97.1%

Table 5: Real inference comparison on POPE (Li et al., 2023b). Experiments adopt 90% pruning ratio.

Method	MME ↑	MMB ↑
w/o Visual Sanitizer	1589.0	51.4
w/o Text-Guided Pruner	1690.0	59.4
w/o Removing Duplicates	1733.3	60.1
w/o Removing High-Norm	1705.6	59.9
w/o High-Norm Aggregation	1737.0	60.2
SinkPruner	**1754.1**	**61.6**

Table 6: **Ablation studies on components.** We report performances on MME((Fu et al., 2023) and MMBench((Liu et al., 2024e)).

SinkPruner exhibits robust cross-modal transferability to videos. To demonstrate the generalizability of *SinkPruner* on video tasks, we evaluate *SinkPruner* on Qwen2.5-VL-7B. As shown in Tab. 3, even under an aggressive **80% pruning ratio**, *SinkPruner* achieves an impressive average performance retention of **98.0%**. It consistently outperforms recent SOTA video pruning methods like DART (Wen et al., 2025) and DivPrune (Alvar et al., 2025), validating its robustness in capturing spatiotemporal cues without any retraining.

We evaluate efficiency on LLaVA-1.5 (Tab. 5). With 90% token pruning, *SinkPruner* reduces total inference time by **33.3%** compared to the full model. While our cascading design introduces a marginal latency overhead compared to competitive methods like VisionZip, it yields a **commanding accuracy advantage** (97.1% vs. 91.8%). Unlike FastV which suffers catastrophic degradation (65.5%), *SinkPruner* achieves the optimal practical trade-off, delivering substantial acceleration without compromising reasoning reliability.

4.3 Ablation Studies

To validate the effectiveness of *SinkPruner*, we dissect the contribution of each component in Tab. 6. The most significant finding is the criticality of the pre-filtering stage: removing the *visual sanitizer* entirely causes a catastrophic performance

Method	Tok	MMB Acc	MMB Rel.	POPE Acc	POPE Rel.
Vanilla (Full)	576	64.7	100%	85.9	100%
VisionZip (Yang et al., 2025b)	64	60.1	92.9%	77.0	89.6%
+ High-Norm Filter	64	**61.4**	**94.9%**	**77.7**	**90.5%**
VisionZip (Yang et al., 2025b)	32	57.7	89.2%	68.7	80.0%
+ High-Norm Filter	32	**59.4**	**91.8%**	**70.4**	**82.0%**

Table 7: **Transferability to existing methods.** Integrating our high-norm filtration module into VisionZip consistently yields performance gains.

drop (*e.g.*, **-10.2%** on MMB). This strongly validates our core premise that without mitigating inherent visual redundancy upstream, the LLM is overwhelmed by attention sinks and dispersion, rendering subsequent reasoning ineffective. Similarly, omitting the *text-guided pruner* leads to a notable decline, confirming that semantic alignment is indispensable for fine-grained understanding. Inside the *visual sanitizer*, the *high-norm removal* proves to be the dominant factor. Retaining these high-norm outlier tokens causes a sharp degradation (MME: $1754.1 \rightarrow 1705.6$), empirically verifying that high-norm outlier tokens are indeed redundant. In contrast, removing similarity-based de-duplication yields a smaller drop, indicating that while spatial diversity matters, the primary gain stems from eliminating the high-norm outlier tokens that disrupt the attention landscape. Notably, *high-norm aggregation* performs slightly better than direct removal, suggesting that preserving a compact summary of these tokens is marginally more effective than discarding them entirely. Additional sensitivity analysis on the high-norm criterion is provided in Appendix B, and Appendix G further ablates the remaining design choices—the pruning layers, the progressive retention schedules, and the salience-to-diversity split—under matched token-computation budgets. Both analyses show flat response surfaces, indicating that our reported configuration is a simple benchmark-independent default rather than a tuned optimum.

4.4 Transferability to Existing Methods

To further demonstrate that our identification of high-norm outlier tokens is a fundamental insight rather than a method-specific heuristic, we evaluate the transferability of our proposed visual filtration module. Specifically, we investigate whether our "high-norm removal" strategy can serve as an orthogonal, plug-and-play enhancement for existing vision-centric pruning frameworks.

We integrate our norm-based separation step into the VisionZip (Yang et al., 2025b) pipeline, applying it before its standard dominant token selection. As shown in Tab. 7, at identical token reduction ratios (32 and 64 tokens), the inclusion of our high-norm outliers filter yields a consistent and significant performance boost ranging from **0.9% to 2.6%** across both MMBench and POPE.

This improvement is particularly noteworthy because VisionZip, as a state-of-the-art attention-based method, inherently prioritizes tokens with high `CLS` *attention*—the exact group we identified as high-norm outlier tokens. By preemptively removing these outliers, we allow the baseline's selection mechanism to focus on truly informative low-norm tokens. These results validate that our proposed filtration module is highly versatile and can be seamlessly generalized to enhance other vision-centric strategies.

5 Conclusion

In this work, we presented *SinkPruner*, a training-free cascading visual token pruning framework for efficient MLLM inference. By filtering high-norm outlier tokens with a *visual sanitizer*, our method suppresses massive activations and attention dispersion in the LLM decoder, thereby enabling more precise *text-guided pruning*.

Extensive experiments on diverse image and video benchmarks demonstrate that *SinkPruner* achieves aggressive token reduction while preserving strong perception and reasoning performance. We further show that our framework generalizes well across different MLLM architectures, from fixed-grid to dynamic-resolution models. Finally, the transferability of the *visual sanitizer* as a plug-and-play module highlights its potential to improve existing pruning methods more broadly.

Limitations. While effective for static images and finite video clips, our current evaluation focuses solely on offline inference with pre-recorded, fixed-length inputs available in advance. Real-world edge applications, such as continuous robotic perception, often require handling potentially infinite video streams with continuously evolving temporal contexts. Adapting our coarse-to-fine pruning strategy to support online streaming—where past visual history is dynamically updated without access to any future knowledge, and where pruning decisions cannot be revisited—remains an open challenge for future work.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, pages 9392–9401. Computer Vision Foundation / IEEE.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023a. Qwen technical report. *CoRR*, abs/2309.16609.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-VL technical report. *CoRR*, abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-VL technical report. *CoRR*, abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *ICLR*. OpenReview.net.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. 2025. Perception encoder: The best visual embeddings are not at the output of the network. *CoRR*, abs/2504.13181.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *NeurIPS*.

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 *ECCV (81)*, volume 15139 of *Lecture Notes in Computer Science*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. 2024b. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2024. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, and 31 others. 2025. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, pages 91–104. Computer Vision Foundation / IEEE.

Peiyan Dong, Mengshu Sun, Alec Lu, Yanyue Xie, Kenneth Liu, Zhenglun Kong, Xin Meng, Zhengang Li, Xue Lin, Zhenman Fang, and Yanzhi Wang. 2023. Heatvit: Hardware-efficient adaptive token pruning for vision transformers. In *HPCA*, pages 442–455. IEEE.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. 2025. Mmtok: Multimodal coverage maximization for efficient inference of vlms. *CoRR*, abs/2508.18264.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2023. EVA: exploring the limits of masked visual representation learning at scale. In *CVPR*, pages 19358–19369. IEEE.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. 2022. Adaptive token sampling for efficient vision transformers. In *ECCV (11)*,

volume 13671 of *Lecture Notes in Computer Science*, pages 396–414. Springer.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. 2023. MME: A comprehensive evaluation benchmark for multimodal large language models. *CoRR*, abs/2306.13394.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, pages 24108–24118. Computer Vision Foundation / IEEE.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven C. H. Hoi. 2023. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *CVPR*, pages 10867–10877. IEEE.

Zi-Yuan Hu, Yanyang Li, Michael R. Lyu, and Liwei Wang. 2023. VL-PET: vision-and-language parameter-efficient tuning via granularity control. In *ICCV*, pages 2998–3008. IEEE.

Zi-Yuan Hu, Yiwu Zhong, Shijia Huang, Michael R. Lyu, and Liwei Wang. 2024. Enhancing temporal modeling of video llms via time gating. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pages 2845–2856. Association for Computational Linguistics.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. OPERA: alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *CVPR*, pages 13418–13427. IEEE.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709. Computer Vision Foundation / IEEE.

Yifan Jiang, Jiarui Zhang, Kexuan Sun, Zhivar Sourati, Kian Ahrabian, Kaixin Ma, Filip Ilievski, and Jay Pujara. 2024. MARVEL: multidimensional abstraction and reasoning through visual evaluation and learning. In *NeurIPS*.

Shibo Jie, Yehui Tang, Ning Ding, Zhi-Hong Deng, Kai Han, and Yunhe Wang. 2024. Memory-space visual prompting for efficient vision-language fine-tuning. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pages 22062–22074. PMLR / OpenReview.net.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 235–251. Springer.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. Openvla: An open-source vision-language-action model. In *CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers. In *KDD*, pages 784–794. ACM.

Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. 2025. Natural language understanding and inference with MLLM in visual question answering: A survey. *ACM Comput. Surv.*, 57(8):190:1–190:36.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2025. LLaVA-OneVision: Easy visual task transfer. *Trans. Mach. Learn. Res.*, 2025.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench: Benchmarking multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13299–13308. IEEE.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023a. BLIP-2: bootstrapping language-image

pre-training with frozen image encoders and large language models. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, pages 22195–22206. IEEE.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024c. Mini-gemini: Mining the potential of multi-modality vision language models. *CoRR*, abs/2403.18814.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *EMNLP*, pages 292–305. Association for Computational Linguistics.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024a. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 5971–5984.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024b. VILA: on pre-training for visual language models. In *CVPR*, pages 26679–26689. IEEE.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2025. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *AAAI*, pages 5334–5342. AAAI Press.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *CVPR*, pages 26286–26296. IEEE.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *NeurIPS*.

Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. 2024c. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *CoRR*, abs/2406.04339.

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

Xiangcheng Liu, Tianyi Wu, and Guodong Guo. 2023b. Adaptive sparse vit: Towards learnable adaptive token pruning by fully exploiting self-attention. In *IJCAI*, pages 1222–1230. ijcai.org.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024e. Mmbench: Is your multi-modal model an all-around player? In *ECCV (6)*, volume 15064 of *Lecture Notes in Computer Science*, pages 216–233. Springer.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, and 6 others. 2025. NVILA: efficient frontier visual language models. In *CVPR*, pages 4122–4134. Computer Vision Foundation / IEEE.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Jie Chen, Zhen Song, and Hairong Zheng. 2026. Apet: Approximation-error guided token compression for efficient vlms. *CoRR*, abs/2602.19870.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL (1)*, pages 12585–12602. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-07-29.

OpenAI. 2025. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`. Accessed: 2025-08-07.

Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. 2025. Mobile edge intelligence for large language models: A contemporary survey. *IEEE Commun. Surv. Tutorials*, 27(6):3820–3860.

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 *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, pages 13937–13949.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, and 34 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*, pages 14313–14323. IEEE.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 22857–22867. IEEE.

Ahmed Sharshar, Latif U Khan, Waseem Ullah, and Mohsen Guizani. 2025. Vision-language models for edge networks: A comprehensive survey. *IEEE Internet of Things Journal*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *CVPR*, pages 8317–8326. Computer Vision Foundation / IEEE.

Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. 2023. Dynamic token pruning in plain vision transformers for semantic segmentation. In *ICCV*, pages 777–786. IEEE.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. In *CVPR*, pages 18992–19001. Computer Vision Foundation / IEEE.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hanrui Wang, Zhekai Zhang, and Song Han. 2021. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *HPCA*, pages 97–110. IEEE.

Hanshi Wang, Yuhao Xu, Zekun Xu, Jin Gao, Yufan Liu, Weiming Hu, Ke Wang, and Zhipeng Zhang. 2025a. Autoprune: Each complexity deserves a pruning policy. *CoRR*, abs/2509.23931.

Hongjie Wang, Bhishma Dedhia, and Niraj K. Jha. 2024. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *CVPR*, pages 16070–16079. IEEE.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025b. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265.

Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. 2023. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *CVPR*, pages 2092–2101. IEEE.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *EMNLP*, pages 9961–9980. Association for Computational Linguistics.

Hao Wu, Yingqi Fan, Jinyang Dai, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. 2026. Hidrop: Hierarchical vision token reduction in mllms via late injection, concave pyramid pruning, and early exit. *CoRR*, abs/2602.23699.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2024. Pyramidrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CoRR*, abs/2410.17247.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. Conical visual concentration for efficient large vision-language models. In *CVPR*, pages 14593–14603. Computer Vision Foundation / IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025a. Qwen3 technical report. *CoRR*, abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025b. Visionzip: Longer is better but not necessary in vision language models. In *CVPR*, pages 19792–19802. Computer Vision Foundation / IEEE.

Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. 2021. TR-BERT: dynamic token reduction for accelerating BERT inference. In *NAACL-HLT*, pages 5798–5809. Association for Computational Linguistics.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2025. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. In *ICLR*. OpenReview.net.

Hongxu Yin, Arash Vahdat, José M. Álvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2022. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, pages 10799–10808. IEEE.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2024. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 57730–57754. PMLR / OpenReview.net.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 9556–9567. IEEE.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952. IEEE.

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

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

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2025c. Llava-video: Video instruction tuning with synthetic data. *Trans. Mach. Learn. Res.*, 2025.

Henry Hengyuan Zhao, Pan Zhou, Difei Gao, Zechen Bai, and Mike Zheng Shou. 2024. LOVA3: learning to visual question answering, asking and assessment. In *NeurIPS*.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. 2025. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. 2024. AIM: adaptive inference of multi-modal llms via token merging and pruning. *CoRR*, abs/2412.03248.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. 2025. Don't just chase "highlighted tokens" in mllms: Revisiting visual holistic context retention. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025*.

In this appendix, we provide further details beyond the main paper, including further validation

of the norm distribution on non-CLIP visual encoders (Sec. A), sensitivity analysis of the high-norm threshold (Sec. B), additional implementation details (Sec. C and Sec. D), additional experimental results on diverse image and video benchmarks (Sec. E), generalization across model scales and visual-token regimes (Sec. F), systematic ablations of the remaining hyperparameters (Sec. G), and additional visualizations of attention outliers and spatial redundancy (Sec. I).

A Experiments on Non-CLIP Visual Encoders

To verify that the high-norm outlier phenomenon is not unique to CLIP-based visual encoders, we further analyze the feature norm distributions of non-CLIP encoder, namely DINOv2. As shown in Fig. 5, both encoders exhibit a clear separation between the majority of regular patch tokens and a small group of high-norm outliers. This suggests that the existence of abnormal high-norm tokens is a broader property of modern vision encoders, rather than an artifact specific to CLIP.

For DINOv2 in particular, although most patch tokens have ℓ_2 norms roughly within the range $[0, 100]$, a small proportion of tokens have substantially larger norms. In our measurement, the fraction of tokens with norm larger than 150 is 2.37%. This confirms that high-norm outliers remain sparse, but sufficiently prominent to distort the feature distribution and potentially interfere with downstream token compression.

At the same time, the exact norm scale differs across encoders. Compared with CLIP-based features used in our main experiments, DINOv2 exhibits a substantially shifted norm range, with its outlier mode appearing at much larger values. This is precisely why *SinkPruner* does not rely on an absolute cutoff. The top-ρ rule of Sec. 3.2 ranks tokens *within* each encoder, so it adapts automatically to a shifted norm range and can be transferred across model families without any per-encoder calibration. An absolute threshold, in contrast, would have to be re-derived from the empirical norm distribution of every new target encoder.

B Sensitivity Analysis

We further study the sensitivity of the high-norm criterion used to identify high-norm outlier tokens in the visual sanitizer, by sweeping an absolute norm threshold τ on LLaVA-1.5.

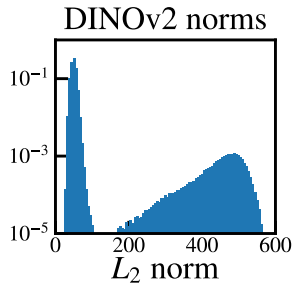


Figure 5: **Norm distributions of non-CLIP visual encoders.** We visualize the patch feature norm distributions of DINO and DINOv2. Both encoders show a small set of high-norm outliers separated from the majority of regular tokens. For DINOv2, although most patch tokens lie below 100, 2.37% of tokens have norms larger than 150, indicating that the high-norm outlier phenomenon generalizes beyond CLIP-based encoders. The differing norm scales motivate our scale-free top-ρ criterion in place of an absolute threshold.

Benchmark	Threshold τ				
	30	45	60	75	90
POPE ↑	10.7	77.8	78.5	78.3	77.5
MME ↑	752	1324	1344	1334	1326

Table 8: **Sensitivity analysis of the high-norm threshold τ.** Performance remains stable across a broad range of threshold values, while an overly small threshold degrades performance.

As shown in Tab. 8, the performance remains stable across a relatively broad range of threshold values once τ is sufficiently large. In particular, choosing τ in the range of $[45, 90]$ leads to only minor variations on both POPE and MME, indicating that our method is not overly sensitive to the exact choice of the threshold. In contrast, an excessively small threshold ($\tau = 30$) causes a clear performance drop, suggesting that overly aggressive filtering may remove useful visual information together with true outliers. The practical message is that high-norm filtering only requires separating the small high-norm tail from the bulk of the distribution, and does not require a precisely tuned operating point. Our reported experiments therefore use the equivalent but scale-free top-ρ rule with $\rho = 1\%$, which realizes this separation without exposing an encoder-dependent constant.

C Experimental Details

C.1 Benchmarks and Metrics

We evaluate our method on a diverse set of widely used benchmarks covering both image-language

and video-language understanding. For **image-based** tasks, we report results on GQA (Hudson and Manning, 2019), MMBench (MMB) and MMB-CN (Liu et al., 2024e), MME (Fu et al., 2023), POPE (Li et al., 2023b), ScienceQA (SQA) (Lu et al., 2022), VQA-v2 (Goyal et al., 2017), and TextVQA (Singh et al., 2019), together with four harder reasoning-oriented benchmarks: MMStar (Chen et al., 2024b), MMMU (Yue et al., 2024), AI2D (Kembhavi et al., 2016), and MM-Vet (Yu et al., 2024). For **video-based** tasks, we additionally evaluate on NextQA (Xiao et al., 2021), MVBench (Li et al., 2024b), SEED-Bench (Li et al., 2024a), and VideoMME (Fu et al., 2025). The detailed information of these tasks is listed below.

GQA (Hudson and Manning, 2019). GQA is a benchmark for real-world visual reasoning and compositional question answering. Its questions are grounded in scene graph structures and paired with functional programs, which makes it well suited for evaluating grounded reasoning, relational understanding, and robustness to superficial language priors.

MMBench / MMB-CN (Liu et al., 2024e). MMBench is a systematically designed multiple-choice benchmark for holistic evaluation of vision-language models. It organizes evaluation into hierarchical ability dimensions spanning perception and reasoning, while MMB-CN provides a Chinese counterpart that enables bilingual and multilingual assessment under a unified protocol.

MME (Fu et al., 2023). MME is a comprehensive benchmark for multimodal large language models that measures both perception and cognition abilities. It consists of 14 subtasks with manually designed instruction-answer pairs and concise prompts, making evaluation more controlled and less sensitive to prompt engineering.

POPE (Li et al., 2023b). POPE focuses on object hallucination in large vision-language models. It reformulates hallucination evaluation as a set of binary probing questions about object presence and reports metrics such as Accuracy, Precision, Recall, and F1 under multiple sampling strategies, providing a direct measure of hallucination behavior.

ScienceQA (Lu et al., 2022). ScienceQA is a multimodal science question answering benchmark with about 21K multiple-choice questions spanning diverse science topics. In addition to answers, it provides lectures and explanations, so it evaluates not only perception but also multi-step scientific reasoning across text and image inputs.

VQA-v2 (Goyal et al., 2017). VQA-v2 is a large-scale open-ended visual question answering benchmark built on diverse real-world images. Each question is paired with ten human answers and evaluated with the standard VQA metric, making it a standard testbed for general visual perception, commonsense grounding, and answer robustness.

TextVQA (Singh et al., 2019). TextVQA evaluates a model's ability to answer questions that require reading and reasoning over scene text embedded in images. Compared with standard VQA, it places greater emphasis on OCR-sensitive perception and fine-grained integration of textual and visual evidence.

MMStar (Chen et al., 2024b). MMStar is a vision-indispensable benchmark curated to reduce language-prior leakage, *i.e.*, questions that can be answered from text alone. It therefore provides a stricter measure of whether a model genuinely grounds its reasoning in the visual input.

MMMU (Yue et al., 2024). MMMU is a massive multi-discipline benchmark of college-level problems spanning science, engineering, and the humanities. It requires knowledge-intensive, expert-level reasoning over heterogeneous visual formats such as charts, chemical structures, and diagrams. We report results on its validation split.

AI2D (Kembhavi et al., 2016). AI2D is a diagram understanding benchmark built from grade-school science diagrams. Answering its questions requires parsing structured visual layouts, including arrows, labels, and part-whole relations, rather than recognizing natural objects.

MM-Vet (Yu et al., 2024). MM-Vet evaluates integrated multimodal capabilities through open-ended questions graded by an LLM judge. Because each question typically requires composing several skills, such as recognition, OCR, spatial reasoning, and knowledge retrieval, it is sensitive to the loss of any single piece of visual evidence.

NextQA (Xiao et al., 2021). NextQA is a video question answering benchmark designed to move beyond descriptive QA toward explaining temporal actions. It emphasizes causal and temporal reasoning in realistic videos, and many questions require tracking events, object interactions, and temporal dependencies across multiple frames.

MVBench (Li et al., 2024b). MVBench is a comprehensive benchmark for video understanding that

targets temporal reasoning abilities of multimodal models. It covers 20 challenging tasks that cannot be solved reliably from a single frame, spanning a broad range of temporal skills from low-level perception to higher-level cognition.

SEED-Bench (Li et al., 2024a). SEED-Bench is a large-scale multiple-choice benchmark for multimodal large language models with accurate human annotations. It spans a broad set of evaluation dimensions and includes both image- and video-centric scenarios, offering an efficient and relatively objective way to compare general multimodal capability.

VideoMME (Fu et al., 2025). VideoMME is a comprehensive benchmark for evaluating multimodal large language models on video analysis. It is designed to assess full-spectrum video understanding, with an emphasis on temporal comprehension, compositional reasoning, and robust performance across diverse real-world video scenarios.

Metrics. We follow the official evaluation protocols of each benchmark. For classification-style benchmarks such as MMBench, we report **accuracy**. For VQA-style benchmarks such as VQA-v2, TextVQA, and NextQA, we report the **official VQA score/accuracy** computed by the released evaluation scripts. For MME, we report the **standard MME score** under its official setup. For POPE, we report **Accuracy**, and additionally Precision/Recall/F1 when required by the benchmark.

C.2 Comparison methods.

We compare *SinkPruner* with existing state-of-the-art (SOTA) visual token reduction methods based on different MLLM architectures (*i.e.*, LLaVA-1.5 (Liu et al., 2024a) and Qwen2.5-VL (Bai et al., 2025b)). These baselines employ diverse strategies—such as token merging, attention-based pruning, adaptive allocation, and hierarchical retention—to improve efficiency by reducing redundant tokens. Specifically, we compare against ToMe (Bolya et al., 2023), MustDrop (Liu et al., 2024d), FastV (Chen et al., 2024a), PDrop (Xing et al., 2024), PruMerge (Shang et al., 2025), SparseVLM (Zhang et al., 2025b), VisionZip (Yang et al., 2025b), Vispruner (Zhang et al., 2025a), DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), HoloV (Zou et al., 2025), MMTok (Dong et al., 2025) and ApET (Ma et al., 2026). Each method offers a unique perspective on balancing computational cost and model performance.

Table 9: **Pruning hyperparameters for LLaVA-1.5-7B under different target token budgets.** The original visual input contains 576 tokens. For low-norm selection, both the salience pool and the diversity pool are set equal to the target retained token budget. The high-norm fraction ρ is shared by all budgets and all backbones.

Target Budget	ρ	Pruning Layers	low-norm Pools	Progressive Token Schedule
Retain-128	1%	$(2, 6, 15)$	$(128, 128)$	$(230, 130, 92)$
Retain-64	1%	$(2, 6, 15)$	$(64, 64)$	$(110, 74, 42)$
Retain-32	1%	$(2, 6, 15)$	$(32, 32)$	$(54, 36, 22)$

For the dynamic-resolution Qwen2.5-VL experiments, VisionZip, DivPrune, and MMTok are re-implemented on top of the same `lmms-eval` harness and the same `max_pixels` setting used for *SinkPruner*, so that all methods observe an identical visual token sequence before pruning.

D Implementation Details

D.1 LLaVA Implementation

Backbone and evaluation. For all experiments on the LLaVA family, we use `LLaVA-1.5-7B` as the base model and follow the official evaluation pipeline released by LLaVA. The original visual input contains a fixed number of 576 image tokens. We adopt the corresponding official toolkit or evaluation server for each benchmark whenever available.

Visual sanitizer. We identify high-norm outliers with the scale-free top-ρ rule, using $\rho = 1\%$ of the visual tokens. For low-norm token selection, we set both the salience pool size and the diversity pool size to match the target retained token budget. Specifically, the two pool sizes are both set to 128 for *Retain-128*, 64 for *Retain-64*, and 32 for *Retain-32*.

Text-guided pruner. For LLaVA-1.5-7B, we perform text-guided pruning at three layers of the language model, namely layers 2, 6, and 15, corresponding to an early, middle, and late stage of multimodal decoding. At each pruning stage, the number of visual tokens is further reduced in a progressive manner. For the three target budgets, the retained visual tokens after the three pruning stages are set to $(230, 130, 92)$, $(110, 74, 42)$, and $(54, 36, 22)$, respectively. This progressive schedule allows the model to gradually remove redundant tokens while preserving important visual evidence for later reasoning.

Pruning configurations. Tab. 9 summarizes the pruning hyperparameters used in our LLaVA experiments.

Table 10: **Pruning hyperparameters for Qwen2.5-VL under different pruning ratios.** The diversity pool is always set to 5% of the current tokens, while the salience pool is adjusted according to the target pruning ratio.

Target Pruning Ratio	Salience Pool	Diversity Pool	Stage-wise Retention Ratios
66.7%	45%	5%	(40%, 35%, 25%)
77.8%	45%	5%	(30%, 20%, 12%)
88.9%	20%	5%	(18%, 11%, 6%)

D.2 Qwen-VL Implementation

Backbone and evaluation. For all experiments on the Qwen family, we use `Qwen2.5-VL` as the base model and conduct evaluation with the `lmms-eval` framework. We use the default frame sampling strategy of the original model. The visual resolution is controlled by setting `max_pixels=376320`.

Visual sanitizer. For Qwen2.5-VL, we adopt the same visual sanitizer as in the LLaVA setting. For low-norm token selection, we divide the candidate tokens into a *salience pool* and a *diversity pool*, following the same terminology used in the main paper. The diversity pool is always set to 5% of the current visual tokens (with at least one token kept), while the salience pool is adjusted according to the target pruning ratio. Specifically, the salience pool is set to 45%, 45%, and 20% of the current visual tokens for the 66.7%, 77.8%, and 88.9% pruning settings, respectively. Since this vision tower provides no [CLS] token, the salience term is instantiated by the CLS-free score described in Sec. D.3.

Text-guided pruner. We apply progressive text-guided pruning in three stages. For different target pruning ratios, the three stages use different retention schedules to gradually reduce the visual tokens. When the target pruning ratio is 66.7%, the three stages retain 40%, 35%, and 25% of the current tokens, respectively. When the target pruning ratio is 77.8%, the corresponding stage-wise retention ratios are 30%, 20%, and 12%. For the most aggressive setting with a target pruning ratio of 88.9%, the three stages retain 18%, 11%, and 6% of the current tokens, respectively. This progressive design enables the model to remove redundant visual tokens step by step while preserving useful evidence for subsequent multimodal reasoning.

Pruning configurations. Tab. 10 summarizes the pruning hyperparameters used in our Qwen2.5-VL experiments.

The pool sizes are defined as proportions rather than fixed counts, since Qwen2.5-VL uses dynamic visual tokenization.

D.3 Salience Scoring for Encoders without a CLS Token

The salience score $\mathcal{A}_{cls}$ used in Sec. 3 presumes a [CLS]-based encoder such as CLIP-ViT, where the attention from the [CLS] token to each patch provides a natural global importance signal. Several modern vision towers, including that of Qwen2.5-VL (Bai et al., 2025b), do not maintain such a token. We emphasize that *SinkPruner* does not fundamentally depend on it: high-norm identification, high-norm aggregation, and diversity-based selection are all defined purely over patch features and are therefore already CLS-free. Only the salience term requires an equivalent substitute.

For non-CLS encoders, we define the salience of visual token j as its average *received* visual self-attention,

$$s_j = \frac{1}{HN} \sum_{h=1}^{H} \sum_{i=1}^{N} A_{ij}^{(h)}, \tag{5}$$

where $A_{ij}^{(h)}$ is the attention weight from visual token i to visual token j in head h, H is the number of attention heads, and N is the number of visual tokens. Intuitively, s_j measures how strongly the remaining visual context attends to token j, so it serves the same purpose as CLS attention: patches that many other patches rely on receive a high score. We compute Eq. equation 5 at the penultimate encoder layer, average over all heads, and substitute s_j for $\mathcal{A}_{cls}$ in the top-k_{res} selection; no other part of the pipeline changes.

Our Qwen2.5-VL experiments validate this substitution. Under 88.9% pruning, *SinkPruner* with the CLS-free score retains 91.8% of the full-model performance, exceeding HoloV by 5.6 points and VisionZip by 4.1 points (Tab. 2). This indicates that the received-attention formulation is a viable drop-in replacement rather than a degraded fallback.

D.4 Batched Diversity Selection

Eq. equation 3 is written as a serial farthest-point rule for clarity: each round appends exactly one token, so obtaining k_{div} diverse tokens requires k_{div} sequential rounds. Because every round depends on the set selected so far, this loop cannot be parallelized on GPU and would dominate the prefill stage when the visual sequence is long, as in high-resolution or video inputs.

Our implementation therefore uses a batched approximation. At round t, let $\mathcal{S}^{(t)}$ be the currently selected set and $\mathcal{R}^{(t)}$ the remaining candidates. We score all candidates simultaneously with a single matrix product, $d(x) = \max_{s \in \mathcal{S}^{(t)}} \mathrm{CosSim}(x, s)$, and append the b least similar ones at once:

$$\mathcal{B}^{(t)} = \mathrm{Bottom}_b\big(\{d(x) \mid x \in \mathcal{R}^{(t)}\}\big), \quad (6)$$

where $\mathrm{Bottom}_b(\cdot)$ returns the b candidates with the smallest scores. The two sets are then updated as $\mathcal{S}^{(t+1)} = \mathcal{S}^{(t)} \cup \mathcal{B}^{(t)}$ and $\mathcal{R}^{(t+1)} = \mathcal{R}^{(t)} \setminus \mathcal{B}^{(t)}$. We use $b = 16$ throughout, which reduces the number of sequential rounds from k_{div} to $\lceil k_{div}/16 \rceil$, *i.e.*, by about $16\times$.

Setting $b = 1$ recovers Eq. equation 3 exactly. For $b > 1$ the two rules differ, because the tokens within one batch are all scored against $\mathcal{S}^{(t)}$ and are therefore not penalized for being similar to each other. The batched variant is thus a close approximation rather than a mathematical equivalent of the serial rule. In practice the gap is small, since candidates are ranked by dissimilarity to an already diverse anchor set $\mathcal{S}^{(t)} \supseteq \mathbf{X}_{res}$, and b is far smaller than $|\mathcal{R}^{(t)}|$. All numbers reported in this paper are produced with the batched variant.

D.5 Compute and Hardware

All experiments are conducted on NVIDIA A800-SXM4-80GB GPUs. Our implementation is based on Python 3.10 with PyTorch 2.1.2 and CUDA 12.1. For all benchmarks, we follow the official evaluation protocols and adopt the default settings of the corresponding baseline implementations.

Attention implementation at pruning layers. The text-guided pruner requires materialized text-to-vision attention scores, which standard FlashAttention kernels do not expose. We therefore fall back to an eager attention implementation at the three pruning layers only; every other layer keeps the efficient kernel and operates on an already shortened visual sequence. The efficiency numbers reported in Tab. 5 and Tab. 13 already include this overhead.

E Additional Experiments

In this section, we provide additional experimental results to complement the main paper. Specifically, we report more comprehensive comparisons on both image and video benchmarks, together with additional real-inference results under different pruning ratios. These results further verify that our method consistently achieves a stronger accuracy-efficiency trade-off across different architectures and evaluation settings.

For image understanding, we present extended benchmark results on LLaVA-1.5 in Tab. 1. For video understanding, we provide additional comparisons on Qwen2.5-VL in Tab. 12. We also include real inference measurements on POPE in Tab. 13, reporting runtime, prefill cost, decoding latency, and relative accuracy under different pruning ratios. Overall, these additional experiments further support the effectiveness and robustness of our method beyond the results shown in the main paper.

F Generalization across Model Scales and Visual-Token Regimes

The main paper evaluates fixed-grid LLaVA-1.5-7B and dynamic-resolution Qwen2.5-VL-7B. This section reports two further settings that stress the two remaining axes of variation: model capacity and visual-sequence length. In both cases the vision encoder, the top-ρ rule, and all other hyperparameters are kept unchanged.

F.1 Transfer across Model Scale

We first run *SinkPruner* on LLaVA-1.5-13B, replacing only the language model. As reported in Tab. 14, the behavior is consistent with the 7B setting. With 192 retained tokens, *SinkPruner* preserves 99.2% of the full-model average and even slightly exceeds the unpruned model on MME and SQA. Under roughly 90% pruning (64 tokens), it preserves 96.2% and outperforms VisionZip by 2.0 relative-performance points, with the largest single-benchmark gap on POPE (84.77 vs. 76.00). This indicates that high-norm outliers are a property of the visual representation rather than of one specific decoder capacity.

F.2 Transfer to High-Resolution Inputs

We next evaluate on LLaVA-NeXT-7B (Liu et al., 2024b), whose any-resolution scheme expands one image into roughly 2,880 visual tokens—five times the 576 tokens of LLaVA-1.5—placing the high-norm observation in a markedly different token regime. All numbers in Tab. 15 are produced by a single end-to-end run of the same evaluation harness, and the **Avg** column normalizes every row against the Upper Bound measured in that run.

SinkPruner attains the best average retention at both budgets, reaching 94.2% with 320 retained

Methods	GQA	MMB	MMB_{CN}	MME	POPE	SQA	VQA_{V2}	VQA_{Text}	Average
Upper Bound, 576 Tokens	61.9	64.7	58.1	1862	85.9	69.5	78.4	58.2	100%
LLaVA-1.5 7B	*Retain 128 Tokens* (Pruning Ratio = 77.8%)								
ToMe (ICLR23)	52.4	53.3	-	1343	62.8	59.6	63.0	49.1	80.4
FastV (ECCV24)	49.6	56.1	56.4	1490	59.6	60.2	61.8	50.6	83.2
MustDrop (2024.11)	56.9	61.1	55.2	1745	78.7	68.5	74.6	56.3	94.6
LLaVA-PruMerge (ICCV25)	53.3	58.1	51.7	1554	67.2	67.1	68.8	54.3	88.0
PDrop (2024.10)	56.0	61.1	56.6	1644	82.3	68.3	72.9	55.1	94.0
VisionZip (CVPR25)	57.6	63.4	56.7	1768	84.7	68.8	75.6	**56.8**	96.9
SparseVLM (ICML25)	56.0	60.0	51.1	1696	80.5	67.1	73.8	54.9	92.6
HoloV (NeurIPS25)	57.7	**63.9**	56.5	1802	84.0	**69.8**	75.5	**56.8**	97.3
ApET (2026.02)	58.9	62.3	56.4	1801	**86.1**	68.7	75.1	53.9	97.0
SinkPruner (Ours)	**59.3**	63.6	**57.9**	**1806**	85.3	**69.8**	**76.8**	55.8	**98.0**
LLaVA-1.5-7B	*Retain 64 Tokens* (Pruning Ratio = 88.9%)								
ToMe (ICLR23)	48.6	43.7	-	1138	52.5	50.0	57.1	45.3	70.1
FastV (ECCV24)	46.1	48.0	52.7	1256	48.0	51.1	55.0	47.8	73.6
MustDrop (2024.11)	53.1	60.0	53.1	1612	68.0	63.4	69.3	54.2	88.5
PDrop (2024.10)	41.9	33.3	50.5	1092	55.9	68.6	69.2	45.9	74.5
VisionZip (CVPR25)	55.1	60.1	55.4	1690	77.0	69.0	72.4	**55.5**	93.1
SparseVLM (ICML25)	52.7	56.2	46.1	1505	75.1	62.2	68.2	51.8	85.6
HoloV (NeurIPS25)	55.3	**63.3**	55.1	1715	80.3	69.5	72.8	55.4	94.5
ApET (2026.02)	56.9	61.2	54.4	1714	**84.4**	68.9	**75.1**	53.9	94.6
SinkPruner (Ours)	**57.4**	62.8	**56.9**	**1754**	83.8	**70.0**	75.0	**55.5**	**96.4**
LLaVA-1.5-7B	*Retain 32 Tokens* (Pruning Ratio = 94.4%)								
ToMe (ICLR23)	43.6	31.6	28.1	828	39.0	41.4	46.8	38.3	55.3
FastV (ECCV24)	41.5	37.8	33.2	885	32.5	42.6	43.4	42.5	57.2
SparseVLM (ICML25)	48.3	51.4	40.6	1047	67.9	57.3	58.6	46.1	74.9
PruMerge+ (2024.05)	51.1	56.8	47.0	941	70.9	68.5	54.9	50.6	80.0
VisionZip (CVPR25)	51.8	57.7	50.3	1247	68.7	68.8	67.1	53.1	85.3
VisPruner (ICCV25)	52.2	58.4	52.7	1271	72.7	69.2	67.7	53.9	87.1
SinkPruner (Ours)	**55.1**	**61.2**	**55.3**	**1363**	**78.5**	**69.9**	**71.4**	**54.8**	**91.2**

Table 11: Comprehensive experimental results on LLaVA-1.5-7B across different image-language benchmarks under different pruning ratios. Best results are in **bold**. Unlike Tab. 1, this table additionally reports VQA-v2, so the averages are computed over eight benchmarks instead of seven and may differ slightly from those in the main paper.

Method	NextQA WUPS	MVBench Acc	SEEDBench Acc	VideoMME Score	Average (%)
Qwen2.5-VL-7B (Full)	26.22	68.10	62.18	60.67	100.0%
DART (EMNLP25)	25.52	65.80	61.00	57.74	96.8%
DivPrune (CVPR25)	25.67	65.85	59.79	57.78	96.5%
SinkPruner (Ours)	**26.03**	**66.70**	**61.80**	**58.59**	**98.3%**

Table 12: Comprehensive experimental results of various methods across different video-language benchmarks under an 80% token pruning ratio. Unlike Tab. 3, this table additionally reports NextQA, so the averages are computed over four benchmarks instead of three and may differ slightly from those in the main paper.

tokens and 92.2% with 160, ahead of the strongest baseline VisPruner by 2.0 and 5.7 points. The margin widens as the budget tightens, which mirrors the trend we observe on LLaVA-1.5: the harder the compression, the more it matters that the retained tokens are not high-norm background. The advantage is concentrated on POPE, where *SinkPruner* loses almost nothing relative to the full model (87.40 vs. 87.61, *i.e.*, 99.8% retention) while every baseline degrades by at least 7 points, and on GQA, where it leads VisPruner by 1.8 and 3.9 points.

TextVQA is the one benchmark on which *SinkPruner* does not lead. It retains 89.0% (320 tokens) and 88.8% (160 tokens) of the full-model score, close to VisionZip (90.1% and 89.1%) but below VisPruner (93.9% and 91.2%). We attribute the residual gap to the interaction between the sanitizer and scene text under tiled high-resolution encoding: small glyph patches can themselves carry large feature norms, so a purely norm-based rule may aggregate a few genuine textual tokens together with background outliers. Notably, the gap does not widen at the tighter budget (2.98 points at 320 tokens versus 1.51 at 160), indicating that the effect is a bounded bias of the criterion rather than a breakdown under aggressive pruning; on LLaVA-1.5 and Qwen2.5-VL, where visual tokens are not tiled, *SinkPruner* gives the best TextVQA scores among all compared methods. Making the sanitizer text-aware, for instance by exempting high-norm tokens with high local edge density, is left to future work.

Methods	Time (mm:ss)	Prefill (ms)	Latency (ms)	Acc. (%)	Time (mm:ss)	Prefill (ms)	Latency (ms)	Acc. (%)
Upper Bound (576 Tokens)	19:28	62.75	121.0	100%	19:28	62.75	121.0	100%
LLaVA-1.5-7B	*Pruning Ratio* = ↓ **77.8%**				*Pruning Ratio* = ↓ **90.0%**			
FastV (ECCV24)	13:56	42.75	83.6	69.4%	12:02	32.2	75.0	65.5%
VisionZip (CVPR25)	13:28	44.11	85.4	95.1%	12:24	36.9	77.4	91.8%
SparseVLM (ICML25)	13:07	46.01	83.5	94.6%	12:20	38.6	77.0	90.5%
SinkPruner (Ours)	14:01	45.24	89.8	99.8%	12:59	37.1	86.0	97.1%

Table 13: Real inference comparison on POPE. Experiments adopt 77.8% and 90% pruning ratios.

Methods	Tok	GQA	MME	POPE	SQA	Avg
Upper Bound	576	63.25	1827.05	87.13	72.73	100.0%
VisionZip (CVPR25)	192	59.60	1770.00	86.40	72.80	97.8%
SinkPruner (Ours)	192	**60.04**	**1839.36**	**87.11**	**73.53**	**99.2%**
VisionZip (CVPR25)	64	56.20	1676.00	76.00	**74.40**	94.2%
SinkPruner (Ours)	64	**57.60**	**1750.13**	**84.77**	73.33	**96.2%**

Table 14: Transfer across model scale on LLaVA-1.5-13B. *SinkPruner* keeps its advantage when the language model grows from 7B to 13B. Best results per budget are in **bold**.

Methods	GQA	VQA_{Text}	POPE	Avg
Upper Bound, 2880 Tokens	64.23	61.37	87.61	100.0%
LLaVA-NeXT-7B	*Retain 320 Tokens* (Pruning Ratio ≈ **88.9%**)			
FastV (ECCV24)	55.90	55.70	71.70	86.5
SparseVLM (ICML25)	56.50	52.40	73.50	85.7
VisionZip (CVPR25)	58.10	55.30	75.00	88.7
VisPruner (ICCV25)	58.40	**57.60**	80.40	92.2
SinkPruner (Ours)	**60.19**	54.62	**87.40**	**94.2**
LLaVA-NeXT-7B	*Retain 160 Tokens* (Pruning Ratio ≈ **94.4%**)			
FastV (ECCV24)	49.80	51.90	51.70	73.7
SparseVLM (ICML25)	50.20	45.10	54.60	71.3
VisionZip (CVPR25)	54.30	54.70	59.40	80.5
VisPruner (ICCV25)	54.70	**56.00**	72.90	86.5
SinkPruner (Ours)	**58.56**	54.49	**84.67**	**92.2**

Table 15: Performance comparison on the high-resolution LLaVA-NeXT-7B, whose visual input expands to about 2,880 tokens. **Avg** is the mean per-benchmark ratio to the Upper Bound. Best results are in **bold**.

G Ablations of the Remaining Hyperparameters

Beyond the high-norm criterion analyzed in Sec. B, *SinkPruner* involves two further groups of design choices: *where* the text-guided pruner acts inside the LLM together with its progressive retention schedule, and *how* the low-norm budget is split between the salience pool and the diversity pool. We stress that these are empirical implementation defaults rather than the core contribution, and that we did not perform a grid search to select them. The analyses below verify that claim.

Pruning layers and progressive schedules. Our default prunes at layers $(2, 6, 15)$, which realizes a coarse-to-fine rule: early layers discard obvious redundancy, middle layers remove cross-modally irrelevant tokens, and later layers retain text-relevant evidence once cross-modal interaction has become discriminative. To test whether performance hinges on this exact triplet, we perturb each pruning layer by ± 1 while keeping the layer-wise token-computation budget $\sum_l n_l$ identical across configurations; the retention schedule is adjusted accordingly so that all seven settings consume the same compute. Tab. 16 shows a flat response surface. POPE accuracy stays within a 1.21-point band (80.17 to 81.38) and MME Total within about 35 points, and no configuration collapses. Notably, our reported setting is *not* the best-performing one, which is consistent with our statement that it was never optimized per benchmark. We note that this analysis establishes insensitivity to pruning *locations* and retention allocation; it does not claim that three stages are theoretically optimal, and we adopt three stages simply as a progressive default that balances gradual pruning against implementation overhead.

Salience-to-diversity split. The visual sanitizer draws representative tokens from a salience pool and a diversity pool. To separate the effect of the *split* from that of the *budget*, we fix the total candidate budget to 2×Retain and vary only the allocation between the two pools. As shown in Tab. 17, POPE accuracy spans 80.49 to 81.91 and MME Total spans 1629.05 to 1682.61, with the two metrics preferring opposite ends of the range: diversity-heavy splits favor POPE while salience-heavy splits favor MME. Rather than picking whichever endpoint maximizes a particular benchmark, we keep the symmetric 5:5 split as a benchmark-independent default.

Config	Prune layers	Retain schedule	POPE Acc.	POPE F1	MME-P	MME-C	MME Total
Default (ours)	$[2, 6, 15]$	$[54, 36, 22]$	80.93	77.52	1333.84	296.79	1630.63
A1: $2 \to 1$	$[1, 6, 15]$	$[56, 36, 22]$	80.69	77.13	1356.92	299.64	1656.56
A2: $2 \to 3$	$[3, 6, 15]$	$[51, 36, 22]$	81.09	77.71	1332.25	301.43	1633.68
B1: $6 \to 5$	$[2, 5, 15]$	$[54, 38, 22]$	80.17	76.20	1333.10	303.93	1637.02
B2: $6 \to 7$	$[2, 7, 15]$	$[54, 34, 22]$	**81.38**	**78.22**	**1358.95**	**306.79**	**1665.74**
C1: $15 \to 14$	$[2, 6, 14]$	$[54, 36, 23]$	81.01	77.64	1348.86	296.43	1645.29
C2: $15 \to 16$	$[2, 6, 16]$	$[54, 36, 21]$	80.90	77.47	1339.32	306.43	1645.75

Table 16: Sensitivity to the choice of pruning layers. Each pruning layer is perturbed by ± 1 while the layer-wise token-computation budget $\sum_l n_l$ is held constant across configurations, so that only the *location* of pruning changes. POPE accuracy varies within a 1.21-point band and MME Total within about 35 points; the configuration used in the paper is not the best-performing one, confirming that it was not selected by benchmark-specific search.

Sal. : Div.	Sal.	Div.	POPE Acc.	POPE F1	MME Total
3 : 7	19	45	81.86	**79.01**	1629.05
4 : 6	26	38	**81.91**	78.93	1662.61
5 : 5 (ours)	32	32	80.93	77.52	1630.63
6 : 4	38	26	80.52	76.90	1675.48
7 : 3	45	19	80.49	76.72	**1682.61**

Table 17: Allocation between the salience pool and the diversity pool under a fixed total budget of $2\times$Retain, which isolates the effect of the split from that of the budget. The response surface is smooth, and we keep the symmetric $5:5$ split as a benchmark-independent default.

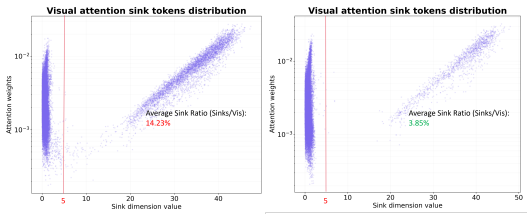


Figure 6: **Visual Attention Sinks in the LLM Decoder.** The scatter plot correlates attention weights (y-axis) with massive activation values (x-axis). **(Left)** Standard text-guided methods (*e.g.*, SparseVLM) operate in a *high-sink regime*, retaining a large cluster of attention sinks (red line, ratio 14.23%). **(Right)** *SinkPruner* operates in a *low-sink regime*. By applying our *visual sanitizer* to reduce inherent visual redundancy, we effectively suppress massive activations, reducing the sink ratio to 3.85% and clarifying the attention landscape.

H Additional Analysis of the Pruning Mechanism

In this section, we provide additional analysis to better understand why *SinkPruner* consistently outperforms prior text-guided pruning methods such as SparseVLM. Our analysis suggests that previous methods are affected by two closely related issues, namely *massive activations* and *text-visual attention dispersion*, both of which are substantially alleviated by our cascading design.

H.1 Mitigating Massive Activations

Existing text-guided methods (*e.g.*, SparseVLM (Zhang et al., 2025b)) are built on the assumption that large attention weights in the LLM decoder indicate semantic relevance. However, recent studies (Kang et al., 2025) show that MLLM decoders often exhibit *massive activations*, where a small set of tokens attract disproportionately large attention regardless of their textual relevance. These tokens behave as *attention sinks*, which can distort token importance estimation and mislead downstream pruning.

Our empirical analysis shows that **reducing inherent visual redundancy significantly alleviates this effect**. As shown in Fig. 6, standard text-guided pruning operates on raw visual sequences and therefore tends to retain many sink tokens due to their inflated attention scores. This leads to a sink ratio of **14.23%**, meaning that a noticeable fraction of the token budget is consumed by non-semantic outliers. In contrast, our *visual sanitizer* first purifies the visual stream before it enters the LLM decoder. This upstream pre-conditioning substantially suppresses the massive activation phenomenon, reducing the sink ratio to **3.85%**. As a result, the following text-guided pruning stage can operate on a much cleaner attention landscape and focus more reliably on truly relevant visual evidence.

H.2 Reducing Text-Visual Attention Dispersion

A second limitation of text-guided pruning methods is *text-visual attention dispersion*. When raw and highly redundant visual tokens are directly fed into

Table 18: **Text-visual attention entropy at the pruning layer.** Lower entropy indicates higher selection confidence.

Method	Entropy (↓)
SparseVLM	6.359
SinkPruner (Ours)	**4.851**

the LLM, cross-modal attention tends to become diffuse and uncertain. We quantify this behavior using Shannon entropy: high entropy indicates that attention is broadly spread and the model lacks discriminative focus, whereas low entropy indicates a sharper and more confident selection pattern.

As shown in Tab. 18, SparseVLM exhibits a relatively high attention entropy of 6.36, reflecting substantial selection uncertainty caused by redundant visual inputs. By contrast, *SinkPruner* significantly reduces the entropy to **4.85**. This result suggests that the *visual sanitizer* acts as an effective denoising stage: by filtering redundant visual content before text-guided pruning, it sharpens the text-to-vision attention distribution and enables the model to identify linguistically relevant tokens with higher confidence.

I Additional Visualizations

In this section, we provide more qualitative visualizations to better illustrate the observations behind our method. These examples offer direct evidence that the problematic attention outliers discussed in the main paper are closely associated with high-norm tokens, and that such tokens are typically concentrated in spatially redundant, non-semantic regions.

I.1 Visualizations of attention outliers and high-norm tokens

Figure 7 presents multiple examples comparing CLS attention maps and feature ℓ_2-norm heatmaps. The results show that sparse attention spikes consistently align with a small set of abnormally high-norm tokens, confirming that these outliers are responsible for the distorted attention patterns observed in the vision encoder. See Fig. 7.

I.2 Visualizations of high-norm patches and low-norm patches

Figure 8 further compares the spatial locations of high-norm and informative low-norm tokens across more examples. We observe that high-norm outliers are mostly concentrated in locally repetitive

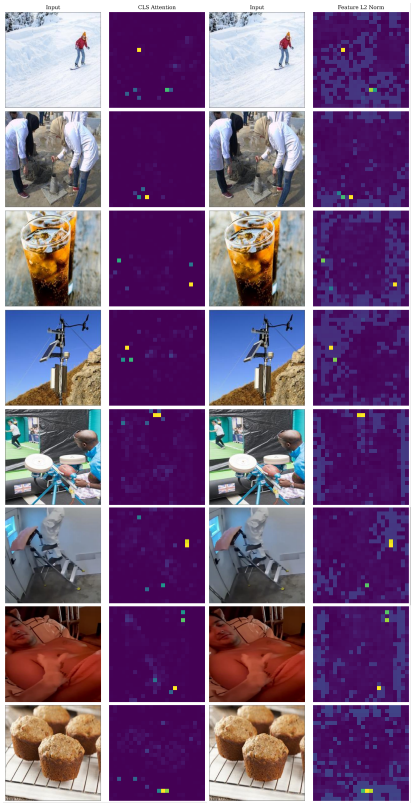


Figure 7: **Attention Artifacts Coincide with High-Norm Outliers.** We show multiple image examples (left to right): *Input*, *CLS Attention*, *Input*, and *Feature ℓ_2 Norm*. The CLS attention maps exhibit sparse, peaky outliers (bright spots) that frequently appear in non-semantic background regions. Crucially, these attention peaks spatially align with abnormally large feature norms in the corresponding ℓ_2-norm heatmaps, indicating that the observed high attention is largely driven by **high-norm outliers** rather than semantic relevance.

background areas, while the retained low-norm tokens tend to lie on visually distinctive and semantically meaningful regions. These visualizations provide additional qualitative support for our claim that high-norm tokens are highly redundant in both spatial and representational dimensions. See Fig. 8.

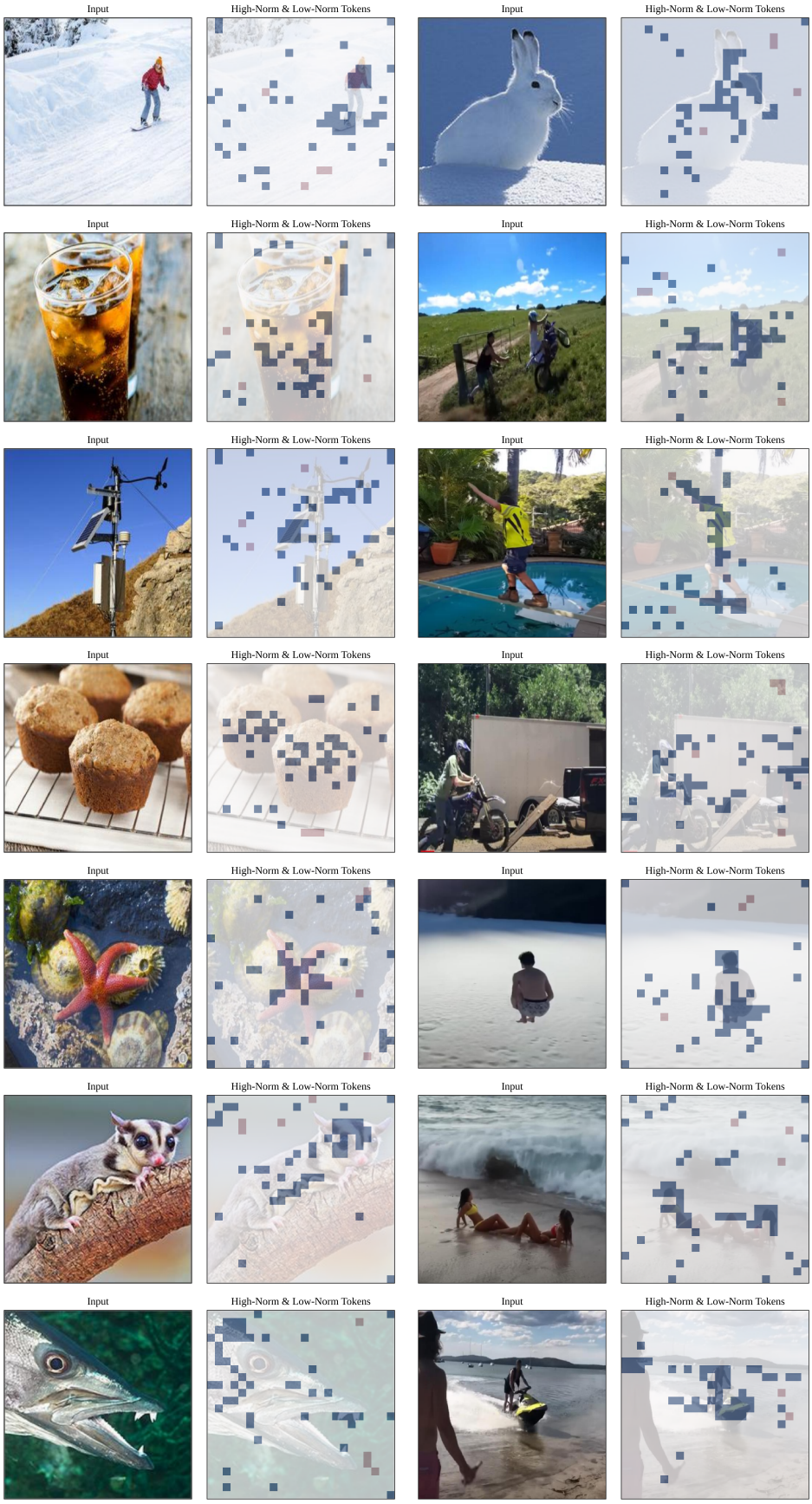


Figure 8: **Spatial Redundancy Analysis.** Visualizations demonstrate that high-norm outlier tokens (red) predominantly appear in non-semantic regions with high local similarity, whereas filtered informative low-norm tokens (blue, top-ranked by `CLS` attention) correspond to unique local features.