arXiv:2602.17196v2 [cs.CV] 6 Sep 2026

EntropyPrune: Matrix Entropy Guided Visual Token Pruning for Multimodal Large Language Models

Yahong Wang[1] **Juncheng Wu**[2] **Zhangkai Ni**[1,†] **Chengmei Yang**[1] **Yihang Liu**[1]
Longzhen Yang[1] **Yuyin Zhou**[2] **Ying Wen**[3] **Lianghua He**[1,4,†]
[1]Tongji University [2]University of California, Santa Cruz
[3]East China Normal University [4]Shanghai Eye Disease Prevention and Treatment Center

Abstract

Multimodal large language models (MLLMs) incur substantial inference cost due to the processing of hundreds of visual tokens per image. Although token pruning has proven effective for accelerating inference, determining when and where to prune remains largely heuristic. Existing approaches typically rely on static, empirically selected layers, which limit interpretability and transferability across models. In this work, we introduce a matrix-entropy perspective and identify an **"Entropy Collapse Layer"** (ECL), where the information content of visual representations exhibits a sharp and consistent drop, which provides a principled criterion for selecting the pruning stage. Building on this observation, we propose **EntropyPrune**, a novel matrix-entropy-guided token pruning framework that quantifies the information value of individual visual tokens and prunes redundant ones without relying on attention maps. Moreover, to enable efficient computation, we exploit the spectral equivalence of dual Gram matrices, reducing the complexity of entropy computation and yielding up to a 64× theoretical speedup. Extensive experiments on diverse multimodal benchmarks demonstrate that EntropyPrune consistently outperforms established competitive pruning methods in both accuracy and efficiency. On LLaVA-1.5-7B, our method achieves a 57.7% reduction in FLOPs while preserving 98.8% of the original performance. Furthermore, EntropyPrune generalizes effectively to high-resolution and video-based models, highlighting the strong robustness and scalability in practical MLLM acceleration [1].

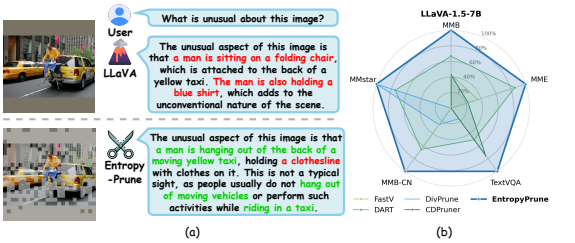


Figure 1: (a) **Comparison between vanilla LLaVA-1.5-7B and EntropyPrune.** Correct answers are highlighted in **green**, while hallucinations are marked in **red**. By removing low-information tokens, EntropyPrune encourages the model to concentrate on more critical details (*e.g.*, the person's state and the car's color). (b) **Performance comparison.** The radial-axis visualization of min-max normalized scores shows that EntropyPrune consistently outperforms competitive methods, including FastV, DART, DivPrune, and CDPruner.

1 Introduction

Multimodal large language models (MLLMs) (Bai et al., 2025; Chen et al., 2025a; Liu et al., 2023; Chen et al., 2024c) have recently achieved remarkable progress on a wide range of visual understanding and reasoning tasks (Lu et al., 2022; Fu et al., 2024; Singh et al., 2019). By integrating a visual encoder with a large language model, these systems enable flexible and general-purpose multimodal reasoning (Chiang et al., 2023; Team, 2023; Bai et al., 2023). However, existing MLLMs tend to represent images using a large number of visual tokens, leading to excessive input sequence lengths and high computational overhead. For example, LLaVA-1.5 (Liu et al., 2024a) represents each image using 576 visual tokens, while Qwen2.5-VL (Bai et al., 2025) adopts a resolution-adaptive strategy that frequently produces several thousand tokens for high-resolution inputs.

To improve the efficiency of MLLM deployment, it is crucial to reduce the number of visual tokens. Existing training-free visual token pruning strategies can be broadly categorized into two groups. Attention-based methods estimate token importance from attention weights and discard tokens with low scores (Chen et al., 2024a; Zhang et al., 2025; Ye et al., 2025; Liu et al., 2024c; Zhao

†Corresponding authors.
[1]Our code is available at https://github.com/YahongWang1/EntropyPrune

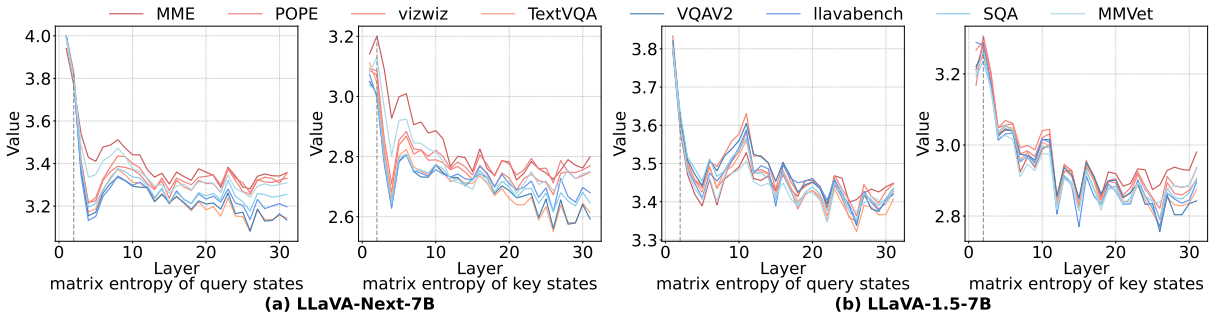


Figure 2: Layer-wise matrix entropy of visual tokens (query and key states) in LLaVA-1.5-7B and LLaVA-Next-7B across eight datasets. A consistent layer-wise trend is observed across different datasets, with a precipitous entropy drop after the second layer.

et al., 2025). Diversity-based methods, in contrast, remove redundant tokens by measuring feature similarity (Bolya et al., 2023; Alvar et al., 2025; Wen et al., 2025; Wang et al., 2025; Zhang et al., 2026). These methods have shown promising results in reducing inference cost while preserving performance. However, a fundamental question remains largely unexplored: **at which layers should pruning be applied?** Most existing methods rely on statically selected pruning layers obtained via empirical tuning or grid search. Such heuristic choices lack interpretability, are model-dependent, and fail to reflect the intrinsic information flow of multimodal representations.

In this work, we revisit token pruning by analyzing the layer-wise information density of MLLMs from an information-theoretic perspective. Inspired by matrix entropy theory (Giraldo et al., 2014), we characterize the information content of visual token representations using the entropy of trace-normalized covariance matrices. This formulation connects naturally to the von Neumann entropy in quantum information theory (von Neumann, 1955), enabling a principled quantification of the informational capacity of visual tokens. We analyze the layer-wise entropy dynamics of visual tokens in LLaVA-1.5-7B and LLaVA-NeXT-7B (Liu et al., 2024b) using randomly sampled inputs from eight datasets, including SQA (Lu et al., 2022), MME (Fu et al., 2024), TextVQA (Singh et al., 2019), POPE (Li et al., 2023), vizwiz (Gurari et al., 2018), llavabench (Liu et al., 2023), MMVet (Yu et al., 2024), and VQAV2 (Goyal et al., 2017).

As illustrated in Figure 2, our analysis reveals a consistent layer-wise pattern of matrix entropy across diverse datasets, independent of query or key states. Here, we define the **"Entropy Collapse Layer"** (ECL) as the layer at which a sharp reduction in matrix entropy is observed. Specifically, the matrix entropy of the query and key states exhibits a precipitous decline after a specific layer (*e.g.*, after the second layer in LLaVA-1.5-7B and LLaVA-Next-7B). This sharp drop indicates a sudden reduction in the information carried by visual tokens, identifying this layer as an interpretable indicator for initiating pruning, unlike prior methods that rely on manual layer selection.

Based on this insight, we propose **EntropyPrune**, a novel training-free token pruning method that leverages entropy collapse to adaptively guide the pruning process. We apply matrix entropy to the token level to precisely quantify the information content of individual visual tokens. Tokens with high entropy are retained, while low-information ones are removed, without relying on attention maps. Nevertheless, naively computing matrix entropy requires eigendecomposition with cubic complexity. To alleviate this bottleneck, we introduce a spectral acceleration strategy that exploits the duality of Gram matrices. This optimization yields a theoretical $64\times$ speedup, making our method computationally feasible in practice. Extensive experiments demonstrate that EntropyPrune substantially reduces computational overhead with negligible impact on model performance. As shown in Figure 1, our method produces more accurate responses with fewer hallucinations than the vanilla model. On LLaVA-1.5-7B, EntropyPrune removes **66.7%** of visual tokens and reduces inference FLOPs by **57.7%**, while retaining **98.8%** of the original performance without any additional training. In addition, EntropyPrune generalizes effectively to long-sequence scenarios, including high-resolution inputs and video understanding, highlighting its robustness and scalability.

In summary, our contributions are as follows:

- We identify a consistent **entropy collapse** phenomenon in MLLMs and introduce the **Entropy Collapse Layer** (ECL) as an inter-

pretable criterion for pruning layer selection.

- We propose **EntropyPrune**, a training-free token pruning framework that ranks visual tokens using matrix entropy and incorporates an efficient spectral acceleration strategy based on dual Gram matrices, achieving a theoretical $64\times$ speedup.
- We conduct extensive evaluations on diverse image and video benchmarks, demonstrating competitive quality–efficiency trade-offs compared with advanced pruning methods.

2 Related Work

2.1 Visual Token Pruning in MLLMs

To alleviate the computational burden introduced by the lengthy visual sequences in MLLMs, visual token pruning has emerged as a promising acceleration strategy. Existing training-free pruning approaches generally fall into two paradigms: *attention-based* and *diversity-based*. **Attention-based methods** treat attention weights as a proxy for token importance, thereby selecting the most important visual tokens (Chen et al., 2024a; Zhang et al., 2025; Zhao et al., 2025; Ju et al., 2024; Xing et al., 2025). However, a critical limitation of attention-based methods is their dependence on explicit attention maps, making them incompatible with FlashAttention (Dao et al., 2022; Dao, 2024). Conversely, **Diversity-based methods** aim to eliminate redundancy by calculating similarity between visual tokens, thereby offering better compatibility with efficient optimizations (Bolya et al., 2023; Alvar et al., 2025; Wen et al., 2025; Zhang et al., 2026; Li et al., 2025). Although these paradigms strike a reasonable trade-off between inference speed and accuracy, they often rely on empirical chosen layers to initiate pruning, which lack interpretability and adaptability.

2.2 Matrix Entropy

Matrix Entropy quantifies the intrinsic information content of data representations directly through the spectral properties of kernel matrices, bypassing explicit probability density estimation (Giraldo et al., 2014). UNComp (Xiong et al., 2025) leverages matrix entropy to quantify the redundancy within the Key-Value (KV) cache of Large Language Models. Most recently, HalfV (Shi et al., 2026) utilizes matrix entropy to analyze cross-modal token interactions across layers and designs a two-stage

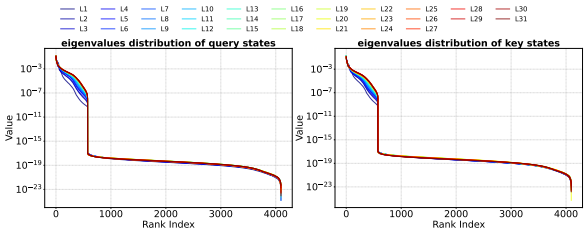


Figure 3: Eigenvalue distribution of the query and key states covariance matrices. Li represents the $i-th$ layer. We visualize the magnitude of eigenvalues across different layers of LLaVA-1.5-7B. The rapid decay observed in the eigenvalues distribution indicates a low-rank structure within the matrices.

pruning framework. Advancing this paradigm, our work identifies a precipitous drop in visual token information at a specific layer, defined as "Entropy Collapse Layer", which generalizes across diverse datasets and MLLMs. Motivated by this discovery, we apply matrix entropy to the token level to precisely quantify the information content of individual visual tokens, enabling targeted pruning at the Entropy Collapse Layer.

3 Methodology

3.1 Preliminaries

We first introduce matrix entropy for characterizing the information content of a visual token sequence. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ denote the visual token matrix, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th token and N is the number of visual tokens. The trace-normalized covariance matrix $\mathbf{\Sigma_X} \in \mathbb{R}^{d\times d}$ is computed as:

$$\mathbf{\Sigma_X} = \frac{1}{N}\sum_{i=1}^{N} \frac{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T}{\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2}, \tag{1}$$

where $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ is the mean vector. This formulation ensures that $\mathrm{tr}(\mathbf{\Sigma_X}) = 1$, fulfilling the requirement for defining matrix entropy. Following (Giraldo et al., 2014), the order-α matrix entropy associated with $\mathbf{\Sigma_X}$ is defined as:

$$S_\alpha(\mathbf{\Sigma_X}) = \frac{1}{1-\alpha}\log(\mathrm{tr}(\mathbf{\Sigma_X^\alpha}))\,. \tag{2}$$

Lemma 1. *Let $\{\sigma_i\}$ be the eigenvalues of $\mathbf{\Sigma_X}$. Then*

$$S_\alpha(\mathbf{\Sigma_X}) = \frac{1}{1-\alpha}\log\left(\sum_i \sigma_i^\alpha\right). \tag{3}$$

The proof is provided in Appendix A.1.

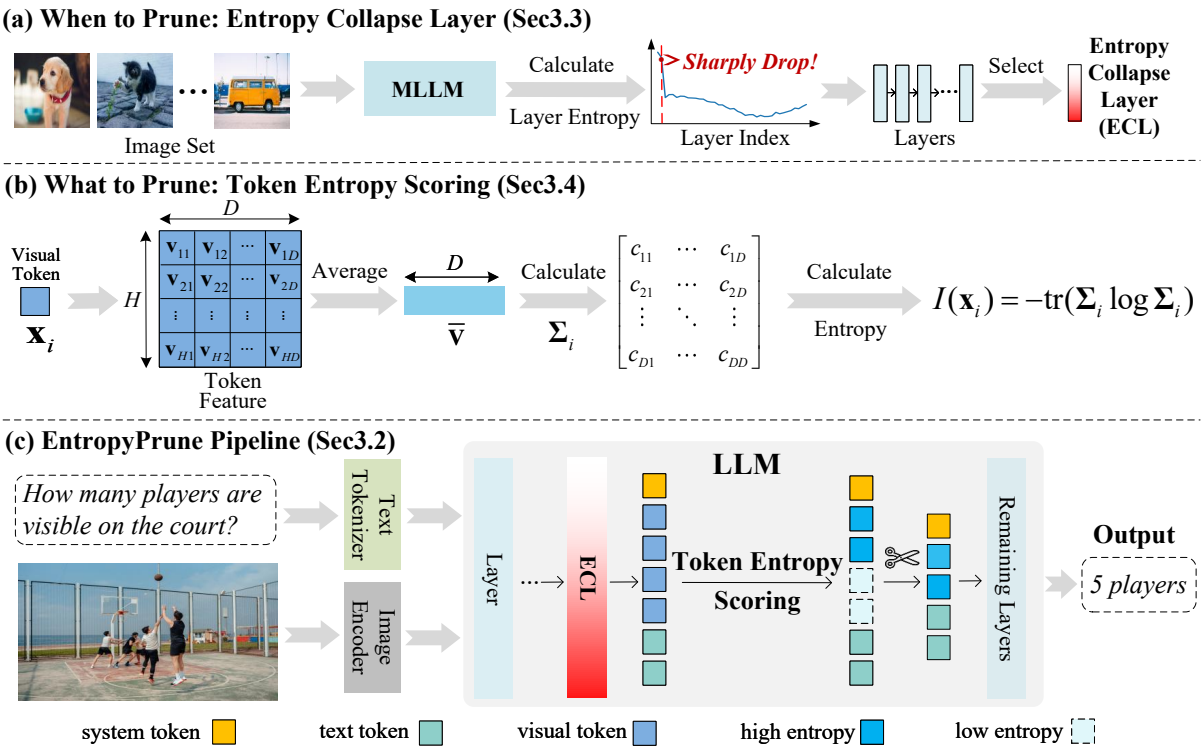


Figure 4: Overview of the proposed EntropyPrune. (a) **When to Prune** identifies the "Entropy Collapse Layer" by detecting a sharp drop in layer-wise matrix entropy. (b) **What to Prune** details the calculation of matrix entropy based on the token's feature. (c) **EntropyPrune Pipeline** illustrates the overall workflow: after reaching "Entropy Collapse Layer", the matrix entropy of each visual token is calculated to guide the pruning of low-entropy ones.

Lemma 2 *As $\alpha \to 1$, the order-1 matrix entropy satisfies:*

$$S(\mathbf{\Sigma_X}) = \lim_{\alpha \to 1} S_\alpha(\mathbf{\Sigma_X}) = -\sum_i \sigma_i \log \sigma_i, \quad (4)$$

where we adopt the convention $0 \log 0 = 0$. The proof of this lemma is provided in Appendix A.2.

Lemma 3. *The trace-normalized covariance matrix $\mathbf{\Sigma_X}$ functions as the density matrix ρ in quantum mechanics, and the order-1 matrix entropy $S(\mathbf{\Sigma_X})$ is mathematically equivalent to the Von Neumann entropy defined in quantum statistical mechanics:*

$$S(\mathbf{\Sigma_X}) \equiv -\mathrm{tr}(\rho \log \rho), \quad (5)$$

The proof is provided in Appendix A.3.

To empirically support this connection, we visualize the eigenvalue distributions of $\mathbf{\Sigma_X}$ computed from the query and key states across layers of LLaVA-1.5-7B. As shown in Figure 3, the spectra are highly concentrated, indicating that most variance (and thus information) is captured by a small number of principal components. Motivated by this observation, we approximate the matrix entropy using the top-k eigenvalues:

$$S(\mathbf{\Sigma_X}) \approx -\sum_{i=1}^{k} \sigma_i \log \sigma_i, \quad (6)$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ are the top-k eigenvalues of $\mathbf{\Sigma_X}$. For brevity, we refer to $S(\mathbf{\Sigma_X})$ as *matrix entropy* in the remainder of this paper.

3.2 EntropyPrune Framework

Figure 4 illustrates the overall pipeline of EntropyPrune. Given an MLLM, we (i) analyze layer-wise matrix entropy to locate the *Entropy Collapse Layer* (ECL), where a sharp information drop indicates increasing redundancy (Figure 4(a)); (ii) quantify the information richness of individual visual tokens via matrix entropy (Figure 4(b)); and (iii) prune low-entropy tokens and forward compact representations for efficient generation (Figure 4(c)).

3.3 When to Prune: Entropy Collapse Layer

As shown in Figure 2, matrix entropy follows a highly consistent depth-wise pattern across eight datasets, despite substantial variation in input images and instructions. This suggests that matrix entropy captures an intrinsic redundancy trend of

multimodal representations, robust to variations in input data. While prior work reports a general information decay with depth (Xing et al., 2025; Lin et al., 2025; Wang et al., 2026), we observe a critical phenomenon within this trend. Specifically, the matrix entropy of query and key states remains high in early layers but drops abruptly after a certain depth (*e.g.*, the second layer for LLaVA-1.5-7B and LLaVA-NeXT-7B). We refer to this transition point as the **Entropy Collapse Layer** (ECL). The collapse indicates rapid compression of visual evidence, after which many tokens become informationally dispensable. Accordingly, ECL serves as an interpretable criterion for selecting the pruning stage, avoiding heuristic layer choices.

3.4 What to Prune: Token Entropy Scoring

Given the pruning stage (*i.e.*, ECL), we rank visual tokens by their information content using token-wise matrix entropy. The feature matrix $\mathbf{M}_i \in \mathbb{R}^{h \times d_h}$ for each token $\mathbf{x}_i$ is given by:

$$\mathbf{M}_i = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_h]^\top, \tag{7}$$

where $\mathbf{v}_j \in \mathbb{R}^{d_h}$ denotes the feature for the j-th attention head and $\bar{\mathbf{v}} = \frac{1}{h}\sum_{j=1}^{h} \mathbf{v}_j$ is the mean feature. Since independent attention heads focus on diverse patterns, constructing $\mathbf{M}_i$ in this manner allows us to evaluate the richness of token information across different representational views. The trace-normalized covariance matrix of the token $\mathbf{x}_i$ is calculated as:

$$\mathbf{\Sigma}_i = \frac{1}{h}\sum_{j=1}^{h} \frac{(\mathbf{v}_j - \bar{\mathbf{v}})(\mathbf{v}_j - \bar{\mathbf{v}})^\top}{\|\mathbf{v}_j - \bar{\mathbf{v}}\|^2}, \tag{8}$$

where $\mathbf{\Sigma}_i \in \mathbb{R}^{d_h \times d_h}$ captures the intra-token correlation structure.

Let $\{\sigma_t\}$ be eigenvalues of $\mathbf{\Sigma}_i$. The token score is:

$$I(\mathbf{x}_i) = -\mathrm{tr}(\mathbf{\Sigma}_i \log \mathbf{\Sigma}_i) = -\sum_{t=1}^{d_h} \sigma_t \log \sigma_t. \tag{9}$$

Higher scores indicate more diverse information distribution, whereas lower scores suggest redundancy. EntropyPrune retains high-score tokens and prunes the rest.

3.5 How to Compute Fast: Spectral Acceleration

Direct eigendecomposition of $\mathbf{\Sigma}_i \in \mathbb{R}^{d_h \times d_h}$ costs $\mathcal{O}(d_h^3)$ time and is expensive in practice. In typical MLLM architectures (*e.g.*, LLaVA-1.5, Qwen2.5-VL), the head dimension d_h is often much larger than the number of heads h. For instance, with $d_h = 128$ and $h = 32$ in LLaVA-1.5-7B, the computational cost becomes prohibitive for real-time inference. To address this issue, we propose a **Spectral Acceleration Strategy** that exploits the identical non-zero spectrum property of dual matrices. Let $\tilde{\mathbf{M}}_i \in \mathbb{R}^{h \times d_h}$ denote the centered matrix of $\mathbf{x}_i$:

$$\tilde{\mathbf{M}}_i = \mathbf{M}_i - \mathbf{1}\bar{\mathbf{v}}^T = [\mathbf{v}_1 - \bar{\mathbf{v}}, \ldots, \mathbf{v}_h - \bar{\mathbf{v}}]^T. \tag{10}$$

To ensure that the resulting covariance matrix is strictly trace-normalized, we perform L_2 normalization on each row of the centered matrix:

$$\tilde{\mathbf{M}}_i \leftarrow \left[\frac{\mathbf{v}_1 - \bar{\mathbf{v}}}{\|\mathbf{v}_1 - \bar{\mathbf{v}}\|_2}, \ldots, \frac{\mathbf{v}_h - \bar{\mathbf{v}}}{\|\mathbf{v}_h - \bar{\mathbf{v}}\|_2}\right]^T. \tag{11}$$

Accordingly, the original traced-normalized covariance matrix can be rewritten as the Gram matrix of the columns of $\tilde{\mathbf{M}}_i$:

$$\mathbf{\Sigma}_i = \frac{1}{h}\tilde{\mathbf{M}}_i^T\tilde{\mathbf{M}}_i \in \mathbb{R}^{d_h \times d_h}. \tag{12}$$

We define its dual counterpart, the Gram matrix of $\mathbf{\Sigma}_i$ as $\tilde{\mathbf{\Sigma}}_i$:

$$\tilde{\mathbf{\Sigma}}_i = \frac{1}{h}\tilde{\mathbf{M}}_i\tilde{\mathbf{M}}_i^T \in \mathbb{R}^{h \times h}. \tag{13}$$

Since $A^\top A$ and $AA^\top$ share identical non-zero eigenvalues, $\mathbf{\Sigma}_i$ and $\tilde{\mathbf{\Sigma}}_i$ have the same spectrum. Therefore, we compute the exact entropy using $\tilde{\mathbf{\Sigma}}_i$:

$$I(\mathbf{x}_i) = -\mathrm{tr}(\tilde{\mathbf{\Sigma}}_i \log \tilde{\mathbf{\Sigma}}_i) = -\sum_{t=1}^{h} \sigma_t \log \sigma_t, \tag{14}$$

reducing the complexity to $\mathcal{O}(h^3)$. For typical settings ($d_h = 128$, $h = 32$), this yields a $64\times$ theoretical speedup.

3.6 Theoretical Analysis of Computational Complexity

In this analysis, we exclusively focus on the LLM backbone of MLLMs, taking LLaVA-1.5-7B as a representative example. As text tokens are typically much shorter than visual tokens, we concentrate on the FLOPs contributed by visual tokens. Let n denote the number of visual tokens, d the hidden size, and m the FFN intermediate size (with SwiGLU), where typically $m \approx \frac{8}{3}d$. For the prefill stage, the

Table 1: **Performance of LLaVA-1.5-7B with EntropyPrune.** The vanilla number of vision tokens is 576. **Acc.** represents the average accuracy across all benchmarks. **Rel.** denotes the relative performance retained compared to the original model. The best performance is highlighted in **bold**.

Method	MMB	MME	VQA^T	MMB^C	MMstar	Acc.(%)	Rel.(%)	FLOPs
Upper Bound, 576 Tokens **(100%)**								
LLaVA-1.5-7B	64.6	1862	58.3	58.1	33.1	56.1	100.0	8.80T
Retain 192 Tokens (↓ **66.7%**)								
FastV (ECCV24)	63.2	1796	56.4	57.4	31.1	54.4	97.0	3.99T
PDrop (CVPR25)	63.3	1765	56.2	56.5	30.7	53.9	96.1	3.79T
DART (EMNLP25)	63.7	1833	57.0	57.1	31.4	54.9	97.9	3.69T
CDPruner (NIPS25)	63.2	1786	57.3	55.6	31.1	54.2	96.6	3.76T
DivPrune (CVPR25)	62.3	1769	56.6	56.2	31.5	54.0	96.1	3.73T
Prumerge (ICCV25)	56.3	1515	50.1	51.3	22.0	46.8	83.3	3.71T
EntropyPrune	**64.4**	**1844**	**57.6**	**57.9**	**31.5**	**55.5**	**98.8**	3.72T

FLOPs per transformer layer can be approximated as:

$$F(n) = 4n^2d + 8nd^2 + 6nmd \approx 4n^2d + 24nd^2. \quad (15)$$

For the detailed derivation, please refer to the Appendix A.4. Suppose the total number of LLM layers is L and the token pruning is applied at the k-th layer. If the token count is reduced by a ratio r ($\hat{n} = (1-r)n$), the total FLOPs reduction ratio is calculated as:

$$R = 1 - \frac{k \cdot F(n) + (L-k) \cdot F(\hat{n})}{L \cdot F(n)}. \quad (16)$$

Given $d \gg n$ in LLaVA-1.5 configurations (*e.g.*, $d = 4096$ vs $n = 576$), the linear term $24nd^2$ dominates the computation over the quadratic attention term $4n^2d$. Thus, $\frac{F(\hat{n})}{F(n)} \approx 1 - r$, and the overall reduction ratio simplifies to $R \approx \frac{L-k}{L} r$.

Furthermore, the additional computational overhead introduced by EntropyPrune is negligible compared to the backbone inference. Specifically, the computational costs for computing the covariance matrix $\tilde{\mathbf{\Sigma}}_i$ and performing the eigenvalue decomposition are $2nhd$ and $4nh^3$, respectively (Golub and Van Loan, 1996). Consequently, the FLOPs of EntropyPrune is approximately $96nd$. Comparing this to a single Transformer layer, the overhead ratio is roughly $\frac{96nd}{4n^2d+24nd^2} \approx \frac{4}{d}$. Given the typically large hidden dimension d, this fraction is vanishingly small, rendering the overhead of EntropyPrune practically negligible.

4 Experiment

4.1 Experiment Setup

Base Models. We adopt the LLaVA series to evaluate our method across different modalities: LLaVA-1.5 (Liu et al., 2024a) (general image), LLaVA-NeXT (Liu et al., 2024b) (high-resolution image), and Video-LLaVA (Lin et al., 2024) (video). Additionally, we incorporate Qwen2.5-VL (Bai et al., 2025) to examine performance on cutting-edge open-source architectures.

Table 2: **Performance of LLaVA-Next-7B with EntropyPrune.** The best performance is highlighted in **bold**, while the second best is underlined.

Method	VQA^T	MMVet	OCR^B	MMMU	AI2D	Acc.
Upper Bound, 2880 Tokens **(100%)**						
LLaVA-Next	61.3	44.0	43.7	28.0	65.6	48.5
Retain 320 Tokens (↓ **88.9%**)						
FastV	49.0	31.1	21.8	29.3	60.2	38.3
PDrop	53.2	29.4	27.1	27.3	61.5	39.7
DART	58.0	39.3	30.5	26.7	62.2	43.3
CDPruner	57.4	38.0	**33.1**	27.3	61.6	43.5
DivPrune	51.4	34.4	29.4	25.3	60.9	40.3
EntropyPrune	**58.5**	**43.8**	30.5	**30.0**	**62.5**	**45.1**

Baselines. We compare our approach with several representative MLLMs token pruning methods, including FastV (Chen et al., 2024a), PDrop (Xing et al., 2025), DART (Wen et al., 2025), DivPrune (Alvar et al., 2025), CDPruner (Zhang et al., 2026), LLaVA-PruMerge(Prumerge) (Shang et al., 2025). More details are provided in Appendix B.2.

4.2 Performance on general token pruning

As shown in Table 1, on the LLaVA-1.5-7B, EntropyPrune consistently surpasses all competing baselines when retaining only 192 visual tokens. For instance, EntropyPrune remarkably achieves 55.5% average accuracy and maintains 98.8% relative performance while removing 66.7% of tokens. Notably, EntropyPrune closely approaches the upper bound performance on benchmarks like MMB and MME while retaining only 192 visual tokens,

Table 3: **Performance of EntropyPrune on Qwen-2.5-VL-7B.**

Method	MMB	MMB^C	MMstar	MMMU	AI2D	Acc.
Upper Bound **(100%)**						
Qwen2.5-VL	79.8	81.0	56.3	58.0	80.8	71.2
Retain 25% Tokens (↓ 75%)						
FastV	77.7	77.2	46.7	42.7	**74.5**	63.8
CDPruner	74.3	77.7	47.8	42.9	71.6	62.9
EntropyPrune	**77.9**	**79.6**	**50.6**	**52.1**	74.1	**66.9**
Retain 12.5% Tokens (↓ 87.5%)						
FastV	53.6	60.0	31.9	38.2	65.5	49.8
CDPruner	71.6	74.7	43.7	42.1	68.3	60.1
EntropyPrune	**73.3**	**75.8**	**44.5**	**48.8**	**69.4**	**62.4**

indicating that our method effectively identifies low-information visual tokens that may otherwise hinder model performance. Overall, EntropyPrune reduces FLOPs by 57.7% , with only a minimal average performance drop of 0.6%.

4.3 Performance on high resolution inputs

To further demonstrate the effectiveness of EntropyPrune when taking high-resolution images as input, we evaluate it on the LLaVA-NeXT-7B, which is capable of processing high resolution images. As shown in Table 2, EntropyPrune demonstrates remarkable scalability, achieving 45.1% average accuracy across five benchmarks while retaining only 11.1% of visual tokens. Notably, EntropyPrune surpasses the base model by 2.0% on MMMU and outperforms the second-best baseline, DART, by 4.5% on MM-Vet. These results underscore the efficacy of EntropyPrune in processing high-resolution inputs.

4.4 EntropyPrune with Qwen architecture

Qwen2.5-VL series adopts a more advanced MLLM architecture that supports Naive Dynamic Resolution to process images of arbitrary aspect ratios. To verify the architectural robustness of EntropyPrune, we conduct experiments on the Qwen2.5-VL-7B. As presented in Table 3, EntropyPrune consistently surpasses competing baselines across different pruning ratios. When retaining 25% of visual tokens, EntropyPrune achieves an average accuracy of 66.9%, outperforming FastV and CDPruner by 3.1% and 4.0%, respectively. Even under aggressive pruning where only 12.5% of tokens are retained, EntropyPrune maintains a robust average accuracy of 62.4%, significantly exceeding FastV (49.8%) and CDPruner (60.1%). Notably, on the challenging MMMU benchmark, EntropyPrune preserves significantly better performance than CDPruner (48.8% vs 42.1%), demonstrating the robust performance across diverse model architectures.

Table 4: **Performance of EntropyPrune on Video-LLaVA-7B.**

Method	MSVD		MSRVTT		Avg.	
	Acc.	Score	Acc.	Score	Acc.	Score
Upper Bound **(100%)**						
Video-LLaVA	53.1	3.2	35.6	2.5	44.4	2.9
Retain 50% Tokens (↓ 50%)						
CDPruner	41.9	2.6	30.1	2.2	36.0	2.4
EntropyPrune	**52.8**	**3.2**	**36.0**	**2.5**	**44.4**	**2.9**

4.5 EntropyPrune with Video tasks

To verify the capability of EntropyPrune in handling video datas, we integrate it with Video-LLaVA-7B and conduct experiments on the MSVD-QA(MSVD) and MSRVTT-QA(MSRVTT) benchmarks. As presented in Table 4, EntropyPrune consistently surpasses CDPruner when retaining 50% of visual tokens. Specifically, it achieves an average accuracy of 44.4%, outperforming CDPruner by a significant margin of 8.4%. Notably, on the MSRVTT benchmark, EntropyPrune achieves 36.0% accuracy, which slightly outperforms the base model (35.6%). These results demonstrates that EntropyPrune effectively eliminates redundant spatiotemporal tokens while preserving the essential visual cues necessary for video reasoning.

4.6 Efficiency Comparison.

To evaluate the efficiency of EntropyPrune, we conduct a comparative analysis against FastV and CDPruner on the LLaVA-1.5-7B. We report prefilling time (Prefill), latency (Lat.), KV cache (KV), GPU memory (Mem.), and FLOPs. The MME benchmark is selected for this evaluation as it encompasses one prefill and one decode stage. As reported in Table 5, when retaining 128 tokens, EntropyPrune achieves 1.6× and 1.4× acceleration in prefilling time and latency, respectively, with only a marginal 4.6% decrease in total score. At this retention level, EntropyPrune reduces KV cache by 77.8% and FLOPs by 67.3%. Compared to FastV and CDPruner, EntropyPrune demonstrates superior efficiency by consuming less GPU memory and achieving faster inference speeds, while maintaining better performance.

4.7 Ablation Study

In this section, we perform ablation studies to validate the two core contributions of our work: the

Table 5: **Efficiency analysis of different pruning methods on LLaVA-1.5-7B.**

Method	Prefill (s)	Lat. (s)	KV (MB)	Mem. (GB)	FLOPs (%)	Score
Upper Bound, 576 Tokens **(100%)**						
LLaVA-1.5-7B	381.8	459.6	288.0	16.2	100	1862
Retain 192 Tokens (↓ **66.7%**)						
FastV	297.5	370.3	96.3	16.0	45.3	1796
CDPruner	360.4	415.7	96.0	19.8	42.7	1786
EntropyPrune	**273.3**	**358.2**	**95.8**	**15.8**	**42.3**	**1844**
Retain 128 Tokens (↓ **77.8%**)						
FastV	256.2	341.1	64.2	15.9	35.7	1735
CDPruner	294.2	352.0	64.0	19.7	33.2	1775
EntropyPrune	**244.3**	**330.1**	**63.9**	**15.7**	**32.7**	**1780**

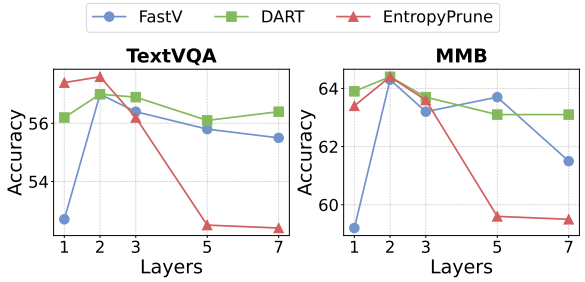


Figure 5: **Ablation study on pruning layer selection.** Experiments are conducted on TextVQA and MMB when retaining 192 tokens. Applying pruning at **Entropy Collapse Layer (Layer 2)** consistently yields the best performance across all methods.

effectiveness of **Entropy Collapse Layer** and our token entropy scoring strategy. All experiments are conducted on the LLaVA-1.5-7B.

4.7.1 Analysis of Pruning Layer Selection

Our analysis about layer-wise entropy posits that information of visual tokens undergoes a precipitous drop at the **Entropy Collapse Layer** (Layer 2 for LLaVA-1.5-7B), identifying it as the optimal stage for token pruning. To empirically verify this hypothesis, we evaluate the performance of EntropyPrune when applied at varying layers, specifically $\{1, 2, 3, 5, 7\}$, while maintaining a consistent budget of retaining 192 tokens on average. For a comprehensive analysis, we further extend this evaluation to FastV and DART, examining whether they exhibit similar layer-sensitivity.

As illustrated in Figure 5, pruning at Layer 2 consistently yields superior performance across all methods compared to earlier or deeper layers. Specifically, EntropyPrune achieves its peak accuracy at Layer 2, achieving 57.6% accuracy on TextVQA and 64.4% accuracy on MMB. It outperforms pruning at Layer 1 and significantly surpasses pruning at deeper layers (e.g., Layer 5), where the accuracy drops sharply to 59.6% on TextVQA and 52.5% on MMB. Similarly, both FastV and DART exhibit their optimal performance at Layer 2. These results strongly validate that the Entropy Collapse Layer serves as both the theoretical and empirical "sweet spot" for visual token pruning.

Table 6: **Ablation study on token selection strategies.**

Method	MMB	MME	SQA	VQA^T	Rel.(%)
Upper Bound, 576 Tokens **(100%)**					
LLaVA-1.5-7B	64.6	1862	69.5	58.3	100.0
Retain 192 Tokens (↓ **66.7%**)					
FastV	64.3	1821	68.9	57.0	98.6
DART	**64.4**	1833	69.0	57.0	98.8
DivPrune	63.2	1795	68.5	57.5	97.9
EntropyPrune	**64.4**	**1844**	**69.1**	**57.6**	**99.2**

4.7.2 Effectiveness of Token Selection Strategy

Building upon the identification of Entropy Collapse Layer as the optimal pruning layer, we evaluate the efficacy of our proposed token selection strategy. We evaluate our method against three baselines: FastV, DART, and DivPrune. For a fair comparison, all methods are configured to prune tokens at the Entropy Collapse Layer (Layer 2), retaining a fixed budget of 192 tokens.

Table 6 presents the comparative results across multiple benchmarks. EntropyPrune consistently outperforms all baselines, achieving the highest relative performance retention of **99.2%**. Notably, it achieves the best scores on MME, SQA, and VQA^T, while tying for the top performance on MMB. This highlights that prioritizing high-entropy visual tokens is vital for maintaining MLLM performance despite significant token reduction.

5 Conclusion

In this paper, we address the significant computational inefficiency of Multimodal Large Language Models (MLLMs) caused by visual token redundancy. Unlike existing pruning approaches that rely on empirical heuristics to select pruning layer, we introduce a rigorous theoretical analysis based on Matrix Entropy. Our analysis reveals the "Entropy Collapse Layer", where visual information drops precipitously, providing a theoretical boundary for determining the optimal pruning layer. Building on this insight, we propose EntropyPrune, a training-free method that selectively prunes tokens based

on their matrix entropy. To ensure efficiency, we incorporate a spectral acceleration strategy using dual Gram matrices, achieving a 64× theoretical speedup in entropy computation. Extensive evaluations demonstrate that EntropyPrune significantly reduces inference FLOPs with minimal degradation in model performance, offering a robust solution for efficient and lightweight MLLM deployment.

Limitations

While our EntropyPrune offers significant inference acceleration for MLLMs, its reliance on white-box architecture access presents a primary constraint. Specifically, calculating the trace-normalized covariance matrices requires direct extraction of intermediate query or key states. Consequently, this methodology is intrinsically incompatible with closed-weight systems, such as the GPT and Claude families, where internal hidden states remain inaccessible via standard APIs.

Ethical considerations

This work introduces EntropyPrune, a training-free token pruning framework designed to optimize the inference efficiency of Multimodal Large Language Models (MLLMs). Our method substantially decreases the computational overhead, carbon footprint, and energy demands typically associated with MLLM deployment. Crucially, because EntropyPrune is entirely training-free, it completely bypasses the massive energy and hardware resource consumption required for model retraining or fine-tuning. Consequently, this approach directly aligns with the objectives of Green AI, offering a highly resource-efficient and environmentally sustainable solution for artificial intelligence application.

Acknowledgement

This research was supported in part by the Yeqisun Joint Funds of the National Natural Science Foundation of China under Grant U2441252, in part by the National Natural Science Foundation of China under Grants 62571372 and 62271155, and in part by the Computational Biology Program (25JS2840100) of Science and Technology Commission of Shanghai Municipality (STCSM).

References

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

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. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your ViT but faster. In *International Conference on Learning Representations*.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*.

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

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? *Preprint*, arXiv:2403.20330.

Yushuo Chen, Tianyi Tang, Erge Xiang, Linjiang Li, Wayne Xin Zhao, Jing Wang, Yunpeng Chai, and Ji-Rong Wen. 2025b. Towards coarse-to-fine evaluation of inference efficiency for large language models. In *Chinese Computational Linguistics: 24th China National Conference, CCL 2025, Jinan, China, August 11–14, 2025, Proceedings*, page 244–264, Berlin, Heidelberg. Springer-Verlag.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198.

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

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 11198–11201, New York, NY, USA. Association for Computing Machinery.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C Principe. 2014. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1):535–548.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, USA.

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)*.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Chen Ju, Haicheng Wang, Haozhe Cheng, Xu Chen, Zhonghua Zhai, Weilin Huang, Jinsong Lan, Shuai Xiao, and Bo Zheng. 2024. Turbo: Informativity-driven acceleration plug-in for vision-language large models. In *European Conference on Computer Vision*, pages 436–455. Springer.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251, Cham. Springer International Publishing.

Duo Li, Zuhao Yang, and Shijian Lu. 2025. Todre: Visual token pruning via diversity and task awareness for efficient large vision-language models. *Preprint*, arXiv:2505.18757.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore. Association for Computational Linguistics.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. 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, Miami, Florida, USA. Association for Computational Linguistics.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2025. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5334–5342.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

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

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

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

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. 2024d. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024e. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12).

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. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Jiaqi Shi, Yuechan Li, Xulong Zhang, Xiaoyang Qu, and Jianzong Wang. 2026. From inheritance to saturation: Disentangling the evolution of visual redundancy for architecture-aware mllm inference acceleration. *Preprint*, arXiv:2604.16462.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM-techreport.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, and Soumya Batra. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

John von Neumann. 1955. *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, Princeton, NJ.

Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, and Enzo Tartaglione. 2025. Folder: Accelerating multi-modal large language models with enhanced performance. *arXiv preprint arXiv:2501.02430*.

Yahong Wang, Juncheng Wu, Zhangkai Ni, Longzhen Yang, Yihang Liu, Chengmei Yang, Ying Wen, Lianghua He, Xianfeng Tang, Hui Liu, and Yuyin Zhou. 2026. When token pruning is worse than random: Understanding visual token information in vllms. *Preprint*, arXiv:2512.07580.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9961–9980, Suzhou, China. Association for Computational Linguistics.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14593–14603.

Jing Xiong, Jianghan Shen, Fanghua Ye, Chaofan Tao, Zhongwei Wan, Jianqiao Lu, Xun Wu, Chuanyang Zheng, Zhijiang Guo, Min Yang, Lingpeng Kong, and Ngai Wong. 2025. UNComp: Can matrix entropy uncover sparsity? — a compressor design from an uncertainty-aware perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4179–4199, Suzhou, China. Association for Computational Linguistics.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*.

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

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22128–22136.

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 *International conference on machine learning*. PMLR.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, 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 *Proceedings of CVPR*.

Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. 2026. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Advances in Neural Information Processing Systems*, 38:25438–25468.

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. 2025. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Zhikai Li, Yibing Song, Kai Wang, Zhangyang Wang, and Yang You. 2025. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19814–19824.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Appendix

A Appendix for Proofs

A.1 Proof of Lemma1

The order-α matrix entropy is defined as:

$$S_\alpha(\mathbf{\Sigma_X}) = \frac{1}{1-\alpha}\log\left(\mathrm{tr}((\mathbf{\Sigma_X})^\alpha)\right). \tag{17}$$

Since $\mathbf{\Sigma_X}$ is the trace-normalized covariance matrix of token matrix $\mathbf{X}$, it is a real symmetric matrix and thus diagonalizable by a unitary matrix $\mathbf{U}$:

$$\mathbf{\Sigma_X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\dagger, \tag{18}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots)$ is the diagonal matrix of the eigenvalues σ_i, $\mathbf{U}$ is the unitary matrix whose columns are the orthonormal eigenvectors of $\mathbf{\Sigma_X}$, and $\mathbf{U}^\dagger$ denotes the conjugate transpose of $\mathbf{U}$.

The α-power of the matrix $\mathbf{\Sigma_X}$ is:

$$(\mathbf{\Sigma_X})^\alpha = (\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\dagger)^\alpha = \mathbf{U}\mathbf{\Lambda}^\alpha\mathbf{U}^\dagger, \tag{19}$$

where $\mathbf{\Lambda}^\alpha$ is also a diagonal matrix with elements σ_i^α:

$$\mathbf{\Lambda}^\alpha = \mathrm{diag}(\sigma_1^\alpha, \sigma_2^\alpha, \ldots). \tag{20}$$

We now calculate the trace $\mathrm{tr}((\mathbf{\Sigma_X})^\alpha)$ using the cyclic property of the trace, $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{CAB})$, and the fact that $\mathbf{U}^\dagger\mathbf{U} = \mathbf{I}$ (since $\mathbf{U}$ is unitary):

$$\begin{aligned}\mathrm{tr}((\mathbf{\Sigma_X})^\alpha) &= \mathrm{tr}(\mathbf{U}\mathbf{\Lambda}^\alpha\mathbf{U}^\dagger)\\ &= \mathrm{tr}(\mathbf{\Lambda}^\alpha\mathbf{U}^\dagger\mathbf{U})\\ &= \mathrm{tr}(\mathbf{\Lambda}^\alpha\mathbf{I})\\ &= \mathrm{tr}(\mathbf{\Lambda}^\alpha).\end{aligned} \tag{21}$$

The trace of the diagonal matrix $\mathbf{\Lambda}^\alpha$ is the sum of its diagonal elements:

$$\mathrm{tr}(\mathbf{\Lambda}^\alpha) = \sum_i \sigma_i^\alpha. \tag{22}$$

Therefore, we get:

$$S_\alpha(\mathbf{\Sigma_X}) = \frac{1}{1-\alpha}\log\left(\sum_i \sigma_i^\alpha\right). \tag{23}$$

A.2 Proof of Lemma 2

To calculate $\lim_{\alpha\to 1}\frac{1}{1-\alpha}\log\left(\sum_i \sigma_i^\alpha\right)$, considering the Taylor expansion of $\sum_i \sigma_i^\alpha$:

$$\begin{aligned}\sum_i \sigma_i^\alpha &= \sum_i \sigma_i \cdot e^{(\alpha-1)\log\sigma_i}\\ &\approx \sum_i \sigma_i\left(1 + (\alpha-1)\log\sigma_i\right)\\ &= 1 + (\alpha-1)\sum_i \sigma_i\log\sigma_i.\end{aligned} \tag{24}$$

As $\alpha \to 1$, we can use the approximation $\log(1+x) \approx x$ for small x. Therefore, we get:

$$S_\alpha(\mathbf{\Sigma_X}) \approx -\sum_i \sigma_i\log\sigma_i. \tag{25}$$

A.3 Proof of Lemma 3

In quantum mechanics, the density matrix ρ serves as a statistical ensemble description of a quantum state, characterizing the inherent uncertainty of the system. We demonstrate that the trace-normalized covariance matrix $\mathbf{\Sigma_X}$ functions as the counterpart to ρ within the visual representation space. Mathematically, $\mathbf{\Sigma_X}$ must be positive semi-definite with a unit trace ($\mathrm{tr}(\rho) = 1$) to serve as a valid density matrix. As a normalized covariance matrix, $\mathbf{\Sigma_X}$ inherently satisfies these constraints: it is symmetric and positive semi-definite by definition, while trace normalization guarantees a unit trace. By definition, $\mathbf{\Sigma_X}$ functions as the statistical description of the feature state, essentially playing the role of the density matrix ρ in quantum state space.

The standard definition of Von Neumann entropy is given by $S = -\mathrm{tr}(\rho\log\rho)$. We substitute ρ with $\mathbf{\Sigma_X}$ and utilize the spectral decomposition. Since $\mathbf{\Sigma_X}$ is a real symmetric matrix, it can be diagonalized as $\mathbf{\Sigma_X} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is an orthogonal matrix satisfying $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and $\mathbf{\Lambda} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots)$ is the diagonal matrix of the eigenvalues σ_i.

The matrix logarithm of $\mathbf{\Sigma_X}$ is defined via its spectral decomposition:

$$\begin{aligned}\log(\mathbf{\Sigma_X}) &= \mathbf{U}\log(\mathbf{\Lambda})\mathbf{U}^T\\ &= \mathbf{U}\mathrm{diag}(\log\sigma_1, \log\sigma_2, \ldots)\mathbf{U}^T.\end{aligned} \tag{26}$$

Substituting this into the trace definition:

$$\begin{aligned}-\mathrm{tr}(\mathbf{\Sigma_X}\log\mathbf{\Sigma_X}) &= -\mathrm{tr}\left((\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)\cdot(\mathbf{U}\log(\mathbf{\Lambda})\mathbf{U}^T)\right)\\ &= -\mathrm{tr}\left(\mathbf{U}\mathbf{\Lambda}(\mathbf{U}^T\mathbf{U})\log(\mathbf{\Lambda})\mathbf{U}^T\right).\end{aligned} \tag{27}$$

Using the property $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, this simplifies to:

$$-\mathrm{tr}\left(\mathbf{U}(\mathbf{\Lambda}\log\mathbf{\Lambda})\mathbf{U}^T\right). \tag{28}$$

By the cyclic property of the trace : $\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{BCA})$, we have:

$$\begin{aligned}-\mathrm{tr}\left(\mathbf{U}(\mathbf{\Lambda}\log\mathbf{\Lambda})\mathbf{U}^T\right) &= -\mathrm{tr}\left((\mathbf{\Lambda}\log\mathbf{\Lambda})\mathbf{U}^T\mathbf{U}\right) \\ &= -\mathrm{tr}(\mathbf{\Lambda}\log\mathbf{\Lambda}). \end{aligned} \tag{29}$$

Since $\mathbf{\Lambda}$ is a diagonal matrix, $\mathbf{\Lambda}\log\mathbf{\Lambda}$ is also diagonal with elements $\sigma_i \log\sigma_i$. The trace is simply the sum of these diagonal elements:

$$-\mathrm{tr}(\mathbf{\Lambda}\log\mathbf{\Lambda}) = -\sum_i \sigma_i \log\sigma_i. \tag{30}$$

A.4 Derivation of FLOPs for LLaVA-1.5-7B

The LLM backbone utilized in LLaVA-1.5-7B is Vicuna-1.5-7B (Zheng et al., 2023), which is obtained by fine-tuning LLaMA-2-7B (Touvron et al., 2023). Its architectural design encompasses two principal components: the multi-head attention block (MHA module) and the feed-forward network (FFN module). Both modules are followed by an RMS normalization (Zhang and Sennrich, 2019) and a residual connection.

The MHA module transforms the input $\boldsymbol{X}$ into query, key, and value matrices $(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ through linear transformations, computes the attention scores, and aggregates the results from multiple heads:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V, \tag{31}$$

$$\begin{aligned}\boldsymbol{O} &= \mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \\ &= \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\intercal}{\sqrt{d}}\right)\boldsymbol{V}, \end{aligned} \tag{32}$$

$$\boldsymbol{X} = \boldsymbol{O}\boldsymbol{W}_O, \tag{33}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_O \in \mathbb{R}^{d\times d}$ denote the learnable parameters, and d represents the hidden dimension.

The FFN module employs the SwiGLU activation function (Shazeer, 2020) to expand the intermediate state dimension via gated linear units, followed by a linear projection to yield the output:

$$\boldsymbol{X} = [\mathrm{Swish}(\boldsymbol{X}\boldsymbol{W}_G) \odot (\boldsymbol{X}\boldsymbol{W}_U)]\boldsymbol{W}_D, \tag{34}$$

where $\odot$ denotes the Hadamard product, while $\boldsymbol{W}_G, \boldsymbol{W}_U \in \mathbb{R}^{d\times m}$ and $\boldsymbol{W}_D \in \mathbb{R}^{m\times d}$ are the parameters, with m representing the intermediate dimension of the FFN.

FLOP Analysis. We analyze the FLOPs per transformer layer during the prefill stage, adopting the counting methodology detailed in (Chen et al., 2025b). For the MHA module, the three linear projections (Equation (31)) involve matrix multiplications requiring $6nd^2$ FLOPs, where n is the sequence length. Applying Rotary Positional Embeddings (RoPE) involves element-wise operations accounting for approximately $6nd$ FLOPs. Regarding the attention mechanism (Equation (32)), computing the correlation between $\boldsymbol{Q}$ and $\boldsymbol{K}$ requires $2n^2d$ FLOPs. The subsequent softmax operation and scaling require roughly $4n^2h$ FLOPs (where h is the number of heads), while the weighted aggregation with matrix $\boldsymbol{V}$ incurs another $2n^2d$ FLOPs. Thus, the core attention mechanism consumes a total of approximately $4n^2d + 4n^2h$ FLOPs. The final output projection (Equation (33)) in the MHA module adds $2nd^2$ FLOPs.

For the FFN module (Equation (34)), the up-projection and gate-projection require $4nmd$ FLOPs. The SwiGLU activation involves element-wise multiplications and the Swish function, contributing $2nm$ FLOPs. The down-projection requires $2nmd$ FLOPs. Additionally, operations for RMS Normalization (RMSNorm) and residual connections account for $5nmd$ FLOPs.

Aggregating these components, the total FLOPs per layer $F(n)$ can be derived as follows:

$$F(n) = \underbrace{4n^2d}_{\text{Attn Score}} + \underbrace{8nd^2}_{\text{Projections}} + \underbrace{6nmd}_{\text{FFN}} + \mathcal{O}(n). \tag{35}$$

Given that the intermediate dimension in LLaMA-2-7B architectures is typically $m \approx \frac{8}{3}d$, the term $6nmd$ approximates to $16nd^2$. Therefore, the total FLOPs can be simplified to:

$$F(n) \approx 4n^2d + 24nd^2. \tag{36}$$

B Detailed Experiment Settings

B.1 Benchmarks

MMBench (**MMB**) (Liu et al., 2024d). MMB is a hierarchical benchmark designed to assess the comprehensive capabilities of MLLMs. It employs a circular evaluation strategy and utilizes ChatGPT to ensure robust matching of model predictions, covering varied abilities such as perception and reasoning.

MMBench-CN (**MMB$^{\mathrm{C}}$**) (Liu et al., 2024d). MMB$^{\mathrm{C}}$ is the Chinese version of MMB.

Table 7: **Performance of LLaVA-1.5-7B with EntropyPrune-q and EntropyPrune-k.** The vanilla number of vision tokens is 576. **Acc.** represents the average accuracy across all benchmarks. **Rel.** denotes the relative performance retained compared to the original model. The best performance is highlighted in **bold**, while the second best is underlined.

Method	MMB	MME	VQA^T	MMB^C	MMstar	Acc.(%)	Rel.(%)	FLOPs
Upper Bound, 576 Tokens **(100%)**								
LLaVA-1.5-7B	64.6	1862	58.3	58.1	33.1	56.1	100.0	8.80T
Retain 192 Tokens (↓ **66.7%**)								
FastV (ECCV24)	63.2	1796	56.4	57.4	31.1	54.4	97.0	3.99T
PDrop (CVPR25)	63.3	1765	56.2	56.5	30.7	53.9	96.1	3.79T
DART (EMNLP25)	63.7	1833	57.0	57.1	31.4	54.9	97.9	3.69T
CDPruner (NIPS25)	63.2	1786	57.3	55.6	31.1	54.2	96.6	3.76T
DivPrune (CVPR25)	62.3	1769	56.6	56.2	31.5	54.0	96.1	3.73T
Prumerge (ICCV25)	56.3	1515	50.1	51.3	22.0	46.8	83.3	3.71T
EntropyPrune-q	**64.4**	**1844**	57.6	**57.9**	31.5	**55.5**	**98.8**	3.72T
EntropyPrune-k	64.3	1837	**57.8**	57.2	**31.7**	55.3	98.6	3.72T

MME (Fu et al., 2024). MME provides a comprehensive evaluation suite consisting of 14 subtasks divided into perception and cognition categories. It is designed to rigorously test models while avoiding common pitfalls like data leakage and prompt engineering sensitivity.

TextVQA (**VQA^T**) (Singh et al., 2019). VQA^T focuses on the integration of text within images, evaluating the model's ability to comprehend and reason about both the visual and textual information present. The benchmark includes a series of visual question-answering tasks where the model must interpret visual content and read embedded text to respond correctly.

MMVet (Yu et al., 2024). MMVet evaluates integrated multimodal capabilities, such as recognition, OCR, and spatial awareness. It proposes an LLM-based evaluator for open-ended outputs.

MMstar (Chen et al., 2024b). MMstar is an elite vision-indispensable benchmark that focuses on "hard" samples. It explicitly filters out questions that can be answered by text alone to ensure that the evaluation genuinely reflects the model's visual understanding capabilities.

AI2D (Kembhavi et al., 2016). AI2D consists of diagrams and corresponding questions, challenging models to parse and reason about diagrammatic structures, arrows, and labels to solve geometry and science problems.

MMMU (Yue et al., 2024). MMMU is a massive multi-discipline multimodal understanding benchmark. It requires expert-level knowledge and reasoning across diverse fields such as art, science, and engineering, testing the breadth and depth of MLLM intelligence.

OCRBench (**OCR^B**) (Liu et al., 2024e). OCR^B is a comprehensive benchmark dedicated to evaluating Optical Character Recognition (OCR) capabilities. It covers various text-related tasks, including text recognition, scene text VQA, and document understanding.

MSVD-QA and MSRVTT-QA (Xu et al., 2017). MSVD-QA and MSRVTT-QA are standard benchmarks for Video Question Answering. They evaluate a model's ability to understand temporal dynamics and reason about events, objects, and actions occurring within video clips.

B.2 Baselines

FastV (Chen et al., 2024a) leverages the attention value of the last text token to rank the information of visual tokens after the third layer.

PDrop (Xing et al., 2025) introduces a pyramid token dropping strategy. It progressively reduces the number of visual tokens as the network depth increases, based on the observation that deep layers in MLLMs often exhibit high redundancy.

DART (Wen et al., 2025) selects a subset of pivot tokens and retains the remaining tokens based on low duplication to the pivots to ensure minimal information loss.

DivPrune (Alvar et al., 2025) utilizes a diversity-based pruning metric. Instead of relying on attention scores, it selects a subset of tokens that maximize the diversity of information, ensuring that the retained tokens cover the most distinct visual features.

CDPruner (Zhang et al., 2026) reformulates token

Table 8: **Comparisons with recent visual token pruning methods on LLaVA-1.5-7B.**

Method	MME	VQA^T	MMStar
Retain 20% Tokens (↓ 80.0%)			
FitPrune	1775	55.7	30.1
EntropyPrune	**1782**	**56.5**	**31.8**
Method	**MME**	**MMB**	**MMStar**
Retain 128 Tokens (↓ 77.8%)			
SparseVLM	1703	60.5	30.7
VisionZip	1756	60.3	29.3
EntropyPrune	**1780**	**60.7**	**31.5**

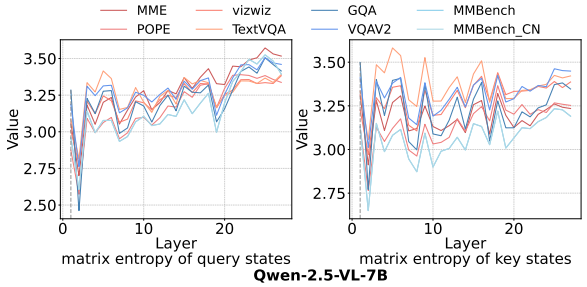


Figure 6: Layer-wise matrix entropy of visual tokens (query and key states) in Qwen-2.5-VL-7B across eight datasets. A consistent entropy collapse is observed after the first layer across different datasets and feature states, demonstrating that the entropy collapse phenomenon also exists in non-LLaVA architectures.

pruning via determinantal point process (DPP) to maximize the conditional diversity of retained tokens based on instruction relevance.

PruMerge (Shang et al., 2025) filters uninformative tokens based on `CLS` attention value and clusters the rest via key similarity.

B.3 Implementation Details.

All experiments are conducted on Nvidia A6000 GPU. The implementation is carried out in Python 3.10, utilizing PyTorch 2.1.2, and CUDA 11.8. All baseline settings follow the original paper.

C Choice of Feature States for Entropy Calculation

EntropyPrune evaluates the information of each visual token based on the matrix entropy defined in Section 3.4. In this section, we compare the performance of two variants: EntropyPrune-q, which utilizes query states for token entropy calculation, and EntropyPrune-k, which utilizes key states. As shown in Table 7, both variants achieve superior performance across various benchmarks, demonstrating the robustness of matrix entropy as a reliable metric for token information. Specifically, when reducing the vision tokens to 192, our two variants rank first and second in average accuracy, with EntropyPrune-q retaining 98.8% performance relative to the base model (near-lossless). For consistency, the final performance reported in Table 1 defaults to using query states.

D Comparisons with Recent Visual Token Pruning Methods

To further evaluate EntropyPrune against recent visual token pruning methods, we additionally compare it with FitPrune (Ye et al., 2025), SparseVLM (Zhang et al., 2025), and VisionZip (Yang et al., 2025) on LLaVA-1.5-7B under matched pruning ratios.

The released implementation of FitPrune supports only a fixed pruning ratio, therefore we compare FitPrune and EntropyPrune under the same 80% pruning ratio. As shown in Table 8, EntropyPrune consistently outperforms FitPrune across all evaluated benchmarks. We further compare EntropyPrune with SparseVLM and VisionZip when retaining 128 of 576 visual tokens, corresponding to a 77.8% pruning ratio. EntropyPrune achieves the best performance across all benchmarks. These results demonstrate that EntropyPrune remains competitive against recent pruning methods under different pruning settings.

The MMBench results of EntropyPrune reported in Table 8 differ slightly from those reported in the main paper. The results in the main paper were obtained using the official MMBench evaluation server together with the LLaVA evaluation code. Since the official evaluation server is no longer available, the additional comparisons reported here are evaluated using VLMEvalKit (Duan et al., 2024) for reproducibility.

E Generality of Entropy Collapse across Architectures

To further examine whether the entropy collapse phenomenon generalizes beyond the LLaVA architecture family, we analyze the layer-wise matrix entropy of visual tokens in Qwen-2.5-VL-7B using randomly sampled inputs from eight datasets, including MME, POPE, VizWiz, TextVQA, GQA, VQAv2, MMBench, and MMBench-CN.

As illustrated in Figure 6, matrix entropy exhibit a highly consistent layer-wise pattern across different datasets, independent of query or key states. Specifically, the matrix entropy sharply decreases

after the first layer. According to our definition, this identifies Layer 1 as the Entropy Collapse Layer (ECL) for Qwen-2.5-VL-7B.

Notably, this behavior closely resembles the entropy dynamics observed in LLaVA-1.5-7B and LLaVA-NeXT-7B, demonstrating that the entropy collapse phenomenon is not specific to the LLaVA architecture family. Moreover, the location of the ECL can vary across different MLLM architectures. These results demonstrate that the entropy collapse is not specific to the LLaVA architecture, but reflects a general phenomenon in the layer-wise evolution of visual token representations in multimodal large language models.