LVPruning: An Effective yet Simple Language-Guided Vision Token Pruning Approach for Multi-modal Large Language Models

Yizheng Sun[1], Yanze Xin[2], Hao Li[1], Jingyuan Sun[1,*], Chenghua Lin[1], Riza Batista-Navarro[1],

[1]University of Manchester, [2]Imperial College London,

***Correspondence:** jingyuan.sun@manchester.ac.uk

Abstract

Multi-modal Large Language Models (MLLMs) have achieved remarkable success by integrating visual and textual modalities. However, they incur significant computational overhead due to the large number of vision tokens processed, limiting their practicality in resource-constrained environments. We introduce Language-Guided Vision Token Pruning (LVPruning) for MLLMs, an effective yet simple method that significantly reduces the computational burden while preserving model performance. LVPruning employs cross-attention modules to compute the importance of vision tokens based on their interaction with language tokens, determining which to prune. Importantly, LVPruning can be integrated without modifying the original MLLM parameters, which makes LVPruning simple to apply or remove. Our experiments show that LVPruning can effectively reduce up to 90% of vision tokens by the middle layer of LLaVA-1.5, resulting in a 62.1% decrease in inference Tera Floating-Point Operations Per Second (TFLOPs), with an average performance loss of just 0.45% across nine multi-modal benchmarks.

1 Introduction

Multi-modal Large Language Models (MLLMs) have achieved impressive results by combining visual and textual information to perform complex tasks that require understanding both modalities (Dai et al., 2023; Li et al., 2023a; Liu et al., 2023a). However, these models can be highly computationally intensive, limiting their practicality in resource-constrained environments (Liu et al., 2023a,b). One important fact that leads to such substantial computational overhead is that these models often process a large number of vision tokens representing input image patches, but not all visual information is equally important for understanding. The human brain, for instance, can focus on salient features while ignoring irrelevant details, allowing for highly efficient visual perception (Treisman, 1988). Inspired by this, there is a growing need to develop MLLMs that can prioritise crucial vision tokens, reducing computational costs without largely sacrificing performance.

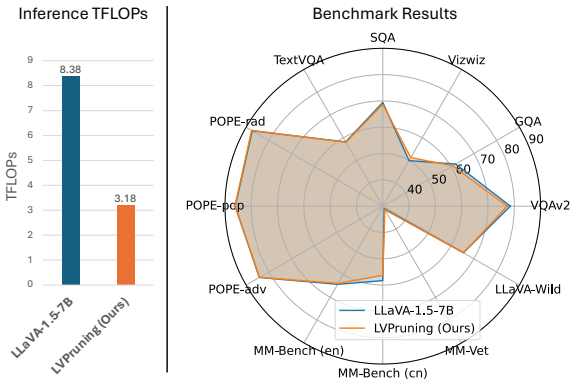

Figure 1: LVPruning can reduce 62.1% of inference TFLOPs for LLaVA-1.5-7B with marginal performance loss across nine multi-modal benchmarks. *All TFLOPs reported in this paper are computed using a dummy input consisting of 1 image and 30 text tokens.

Previous approaches for enhancing the computational efficiency of MLLMs have explored various strategies. Models utilising Q-former as the vision encoder condense visual information into a smaller set of tokens. Though effectively reducing computational load, such condensation potentially leads to a loss of essential visual information, compromising performance (Li et al., 2023a; Dai et al., 2023; Zhu et al., 2024). On the other hand, models such as LLaVA pass all vision tokens through a simple Multi-Layer Perceptron (MLP) connector to the language model, achieving high performance but at the cost of increased computational demands (Liu et al., 2023a,b). Additionally, token compression techniques (Rao et al., 2021; Bolya et al., 2023; Chen et al., 2023) that detect important vision tokens solely based on visual features have shown

Findings of the Association for Computational Linguistics: NAACL 2025, pages 4299–4308
April 29 - May 4, 2025

promise in single-modal tasks but cannot make full use of the interaction between visual and linguistic information in MLLMs. These highlight a trade-off between computational efficiency and model performance, indicating a need for solutions that can balance both aspects effectively and efficiently.

To address these challenges, we propose Language-Guided Vision Token Pruning (LVPruning), a simple yet effective method that dynamically reduces the number of vision tokens in MLLMs based on their relevance to the language context. We introduce lightweight cross-attention decision modules where vision tokens attend to language tokens to compute importance scores. This relevance scoring allows the model to decide whether to keep or prune each vision token, effectively filtering out less informative visual data. By integrating these decision modules into various layers of the MLLM, LVPruning enables progressive token pruning as the model processes deeper layers. During training, we freeze all original model parameters and only train the inserted decision modules, ensuring that the base model remains unchanged and the pruning mechanism can be easily applied or removed.

Our contributions are threefold. First, as shown in Figure 1, we demonstrate that LVPruning can significantly reduce computational costs—up to a 62.1% decrease in inference TFLOPs—by pruning as much as 90% of vision tokens without substantially affecting model performance. Second, we introduce a novel, language-guided token pruning mechanism that is both effective and easy to integrate into existing MLLMs, requiring minimal changes to the original architecture. Third, our method allows for adjustable token pruning ratios during inference without retraining, offering flexibility in balancing efficiency and performance according to specific needs. Through extensive experiments on various multi-modal benchmarks, we show that LVPruning provides a practical solution to enhance the efficiency of MLLMs while maintaining their ability to understand and generate accurate multi-modal content.

2 Related Work

Multi-modal Large Language Models: Recent advancements in MLLMs have significantly enhanced the integration of visual and textual modalities. BLIP-2 (Li et al., 2023a) introduced a two-stage learning framework that connects pre-trained vision models with language models using a Q-former model as a vision encoder, effectively generating a condensed set of vision tokens for efficient processing. Building upon BLIP-2, InstructBLIP (Dai et al., 2023) and MiniGPT-4 (Zhu et al., 2024) incorporated instruction tuning to improve the model's ability to follow complex prompts and perform diverse tasks. Alternatively, models such as LLaVA-1.5 (Liu et al., 2023a) directly input all vision tokens from pre-trained vision encoders into the language model. While this approach achieves higher performance owing to richer visual information, it results in substantial computational overhead. These models exemplify the trade-off between computational efficiency and performance, underscoring the need for approaches that can balance both aspects without compromising accuracy.

Efficient Transformers: Many techniques have been proposed to improve computation efficiency for transformer models, such as knowledge distillation (Hinton et al., 2015), token merging/pruning (Bolya et al., 2023; Rao et al., 2021; Chen et al., 2023), and quantisation (Gong et al., 2014; Wang et al., 2019). For Natural Language Processing (NLP) tasks, methods like DistilBERT (Sanh et al., 2019) and MiniLM (Wang et al., 2020) use knowledge distillation to create smaller models for more efficient inference. For computer vision tasks, Liang et al. (2022); Bolya et al. (2023) and Chen et al. (2023) focus on pruning or merging tokens based on their importance in image classification. These approaches reduce the number of vision tokens by identifying less informative patches or merging similar tokens during inference. The closest work to ours is DynamicVit (Rao et al., 2021), which uses MLP layers to predict token pruning decisions. They hierarchically insert multiple pruning layers into vision transformer-based models for the image classification task. However, these methods are specifically designed for single-modal targets and do not address challenges in multi-modal settings. Our research distinguishes itself by focusing on token pruning for MLLMs in the context of image comprehension tasks.

3 Methodology

In this section, we present the LVPruning framework in detail. As shown in Figure 2, LVPruning is designed for MLLMs that pass all vision tokens through an MLP connector into the language model. The architecture consists of a transformer-based pre-trained CLIP vision encoder, an MLP vision-

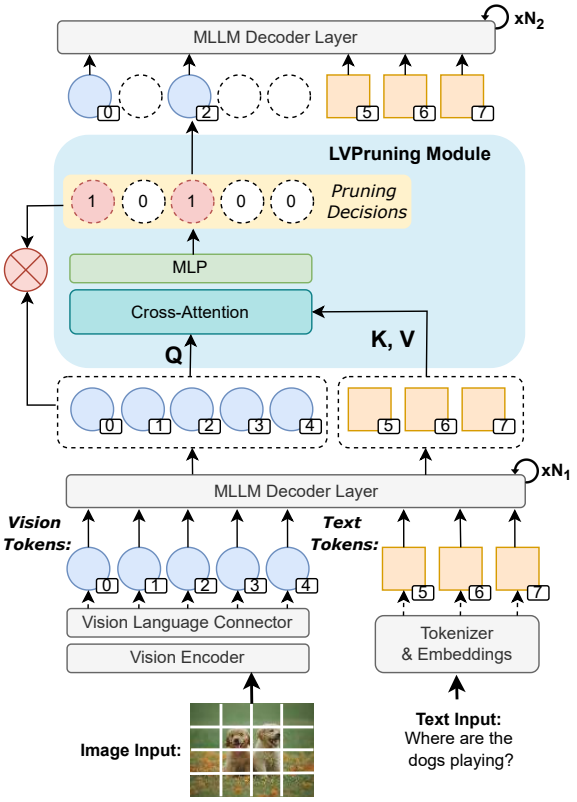

Figure 2: Overall Framework Architecture. LVPruning modules are incorporated into specific layers of an MLLM, where vision tokens serve as queries and language tokens act as keys and values. A pruning decision is predicted for each vision token. The operation denoted by $\otimes$ applies these decisions—serving as attention masking during training and token removal via indexing during inference.

language connector, and an LLM backbone. First, an image input is divided into patches and processed by the CLIP model (Radford et al., 2021), such that each image patch becomes a representative vision token. Next, with the vision language connector projecting vision tokens into the dimension of the LLM's text space, the concatenated vision and text tokens are fed into the LLM for causal text generation. In specific layers of the LLM, cross-attention decision modules dynamically select the most salient token (the one with the highest attention score) to guide inference, removing redundant tokens. During training, we apply attention masks (Rao et al., 2021) to mask out pruned vision tokens. Importantly, instead of updating positional embeddings after token pruning, we retain the original positional embeddings for the remaining vision tokens, as used in standard LLMs (Touvron et al., 2023).

3.1 Cross-Attention Decision Module

We now describe the detailed architecture of the cross-attention decision module designed for token pruning. A decision module, which is responsible for selecting and discarding vision tokens, comprises cross-attention layers and an MLP layer. Multiple instances of these modules are inserted into different layers of the LLM backbone for progressive pruning. Let $\mathbf{H} \in \mathbb{R}^{N \times d}$ represent the output from an LLM hidden layer, where $\boldsymbol{N}$ is the sequence length and $\boldsymbol{d}$ is the dimension of the hidden representations. $\mathbf{H}$ contains the subset of vision tokens and text tokens. We define the set of the vision tokens indices as $\mathbf{I_V} = \{n_{v_i} \mid v_i \in \mathbb{N}, 0 \leq v_i < N\}$, and the set of text token indices as $\mathbf{I_T} = \{n_{t_i} \mid t_i \in \mathbb{N}, (0 < t_i \leq N) \wedge (t_i \notin I_V)\}$. We use the vision tokens as the query tokens

$$\mathbf{Q} = W_q H_{I_V} \in \mathbb{R}^{|I_V| \times d}, \tag{1}$$

and text tokens as the key and value tokens

$$\mathbf{K}, \mathbf{V} = (W_k / W_v) H_{I_T} \in \mathbb{R}^{|I_T| \times d}, \tag{2}$$

where W_q, W_k, W_v are linear projection layers. We then compute the attention matrix and feed the output to a feed-forward network (FFN), as described by Vaswani et al. (2017).

$$\mathbf{O} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V + Q. \tag{3}$$

Inspired by Rao et al. (2021), we feed the output from the FFN to a linear layer W_O to predict scores of keeping and removing vision tokens:

$$\boldsymbol{\gamma} = W_O O \in \mathbb{R}^{|I_V| \times 2}, \tag{4}$$

where $\boldsymbol{\gamma_{i,0}}$ represents the score for keeping the vision token $\mathbf{H_{V,i}}$ and $\boldsymbol{\gamma_{i,1}}$ represents the score for removing the vision token $\mathbf{H_{V,i}}$. The decisions of vision token pruning are then generated based on $\boldsymbol{\gamma}$. The mechanism for generating and applying decisions differs between training and inference. Multiple decision modules are inserted into different layers of the LLM, such that the vision tokens are pruned progressively throughout the LLM.

3.2 End-to-End Training

To ensure that the process of generating and applying token pruning decisions based on $\boldsymbol{\gamma}$ is differentiable, where $\boldsymbol{\gamma}$ denoted as the output from each cross-attention decision module, we draw inspiration from Rao et al. (2021). Specifically, we

apply the Gumbel-Softmax distribution to $\boldsymbol{\gamma}$, redistributing it into one-hot vectors $\boldsymbol{D^{GS}}$. The first dimension of each vector is then used as the decision $\boldsymbol{D}$, determining whether to retain a given token:

$$\boldsymbol{D^{GS}} = \text{Gumbel-Softmax}(\gamma) \in \{0,1\}^{|I_V| \times 2}, \tag{5}$$

$$\boldsymbol{D} = D^{GS}_{:,0} \in \{0,1\}^{|I_V| \times 1}, \tag{6}$$

where $\boldsymbol{D_i} = \mathbf{1}$ means that the vision token $\mathbf{H_{I_{V,i}}}$ is kept; Otherwise it's removed. The number of kept tokens determined by the decision process is not fixed during training, and directly removing unwanted vision tokens will impede batch processing. To address this concern, we construct attention masks $\boldsymbol{M}$ based on $\boldsymbol{D}$ for both vision and language tokens. Specifically,

$$\boldsymbol{M_{i,j}} = \begin{cases} 1 & \text{if } i = j \text{ or } j \in I_T \\ D_j & \text{if } i \neq j \text{ and } j \in I_V \end{cases}, \quad 1 \leq i, j \leq N. \tag{7}$$

Therefore, $\boldsymbol{M}$ is constructed such that all language tokens are assigned a mask value of 1, while the vision tokens are masked according to the token pruning decisions $\boldsymbol{D}$. Additionally, all diagonal elements of $\boldsymbol{M}$ are set to 1 to improve numerical stability.

However, $\boldsymbol{M}$ disregards the original causal and padding attention masks. To address this, we first apply the original attention mask $\bar{\boldsymbol{M}}$ to the raw attention scores to obtain causal attention matrix $\bar{\boldsymbol{A}}$. Then, we apply $\boldsymbol{M}$ with the $Softmax$ operation to $\bar{\boldsymbol{A}}$ to get the final attention matrix $\hat{\boldsymbol{A}}$. Specifically, we define $\boldsymbol{M_l}$ as the attention mask generated from the decision module at layer $\boldsymbol{l}$. The attention scores at layer $\boldsymbol{l + x}$ is calculated by

$$\hat{\boldsymbol{A}}_{\boldsymbol{l+x}} = \hat{Softmax}(\bar{A}_{l+x}, M_l), \tag{8}$$

$$\hat{Softmax}(A, M) = \frac{\exp(A_{i,j})M_{i,j}}{\sum_{k=1}^{N} \exp(A_{i,k})M_{i,k}}, \quad 1 \leq i, j \leq N, \tag{9}$$

where $(\boldsymbol{l + x}) \in \mathbb{N} < l'$. $\boldsymbol{l'}$ is the position layer of the next decision module. If $M^L_{i,j} = 0$, the attention score for token H_j will be 0 in the final attention matrix, resulting in H_j not contributing to any other tokens. In addition, we define $\boldsymbol{D_l}, \boldsymbol{D_{l'}}$ as the token pruning decisions obtained from layer l, l', respectively and $l' > l$. We update $\boldsymbol{D_{l'}}$ by

$$\boldsymbol{D_{l'}} \leftarrow D_l \odot D_{l'}, \tag{10}$$

where $\odot$ is element-wise production, which means a previously removed vision token will never be used again.

In summary, Equations 7 - 9 remove the effects of unwanted vision tokens on other tokens while keeping the number of total tokens unchanged. With Equations 5, 6, 8 and 9, the process of generating and applying token pruning decisions becomes fully differentiable. These two factors facilitate the end-to-end training capability of LVPruning.

Training Objectives: The training objectives of LVPruning are designed to teach the decision modules to remove vision tokens according to predetermined ratios at different layers while fine-tuning the MLLM to maintain its vision instruction-following capability despite the token pruning. The primary training objective is causal language modelling for instruction tuning, as described by Vaswani et al. (2017); Liu et al. (2023b); Touvron et al. (2023). Since causal language modelling is a widely used loss function, we do not formally define it in this paper, referring to it as $\boldsymbol{\mathcal{L}_{causal}}$.

Additionally, to ensure that the ratio of retained vision tokens aligns with predefined values at each decision module, we insert $\boldsymbol{S}$ decision modules into the LLM at specific layer indices $\boldsymbol{L_{idx}} = [l_1, ..., l_S]$, with target token retention ratios $\mathbf{P} = [\rho_1, ..., \rho_S]$. To enforce this, we apply Mean Squared Error (MSE) loss $\mathcal{L}_{\text{ratio}}$ to constrain the token pruning decisions:

$$\mathcal{L}_{\text{ratio}} = \frac{1}{S} \sum_{s=1}^{S} (\rho_s - \frac{1}{|I_V|} \sum_{i=1}^{|I_V|} D_{l_s,i})^2, \tag{11}$$

where $\delta(D_{l_s}, \rho_s)$ is the Huber loss and β is a threshold that determines the loss function used. We set $\beta = 0.5$ in all our experiments. The final training objective is the weighted sum of $\boldsymbol{\mathcal{L}_{causal}}$ and $\boldsymbol{\mathcal{L}}_{\text{ratio}}$:

$$\boldsymbol{\mathcal{L}} = \lambda_{causal}\mathcal{L}_{causal} + \lambda_{ratio}\mathcal{L}_{ratio}. \tag{12}$$

3.3 Inference

During the training phase, attention masks are employed to exclude the impact of irrelevant vision tokens. However, during inference, it is necessary to remove these tokens to reduce computational costs, which introduce significant practical difficulties. First, the quantity of retained vision tokens, as determined by $\boldsymbol{D}$, is variable, thereby complicating the process of batch inference. Second, contemporary LLMs generally utilise positional embeddings

for tokens at each layer. It is essential to maintain the original positional embeddings for the retained tokens to ensure alignment with the distribution seen during training.

To overcome the first issue, we define a set of token kept ratios $\hat{\mathbf{P}} = [\hat{\rho_1}, ..., \hat{\rho_S}]$ during inference. Note that the inference ratios do not have to be the same as the training ratios. At the s-th token pruning layer, we first sort the decision scores

$$\mathbf{Q^s} = argsort(D_{ls}). \tag{13}$$

We then keep the top $\boldsymbol{k_s} = \rho_s \times \mid I_V \mid$ vision tokens with the highest scores. The kept vision token indices among all vision tokens are $\hat{\boldsymbol{I}}^{\boldsymbol{s}} = \{Q^s_{1:k_s}\}$, and the kept vision token indices among all tokens are $\hat{\boldsymbol{I}}^{\boldsymbol{s}}_{\boldsymbol{v}} = I_{v,\hat{I}^s}$. We define $\mathbf{PE^1}$ as the positional embeddings at the first layer. To ensure that the positional embeddings for both vision and text tokens remain unchanged after each pruning, the positional embeddings at the s-th token pruning layer are defined as

$$\boldsymbol{PE^s} = [PE^1_{\hat{I}^s_v}, PE^1_{I_T}]. \tag{14}$$

4 Experimental Setup

The objective of our experiments is to investigate the feasibility of employing token pruning techniques to enhance the efficiency of MLLMs. Specifically, we ask: *(i)* Does LVPruning effectively reduce computational costs while keeping the performance unchanged and to what extent can vision tokens be pruned? *(ii)* Compared with state-of-the-art MLLMs, does LVPruning achieve a balance between computational costs and model performance? To answer question *(i)* we compare LVPruning with the base MLLM on various benchmarks using different ratios of kept vision tokens. To answer *(ii)*, we compare the relationship between inference TFLOPs and model performance for LVPruning and various state-of-the-art MLLMs.

4.1 Implementation Details

In all our experiments, we apply LVPruning to LLaVA-1.5-7B (Liu et al., 2023a) (hereafter referred as LVPruning) by inserting $S = 3$ decision modules with token kept ratio $\mathbf{P} = [\rho, \rho - 0.2, \rho - 0.4]$, where $\rho = 0.5$ for training. These modules are inserted after the 1st, 8th, and 16th layers of LLaMA LLM. In each token pruning layer, we utilise 2 sequential cross-attention blocks, each with 8 attention heads. The FFNs in these cross-attention blocks follow the architecture: [LayerNorm, Linear(C, 2C), SiLu activation, Linear(2C, C), LayerNorm]. We follow most of LLaVA-1.5's training settings, freezing all parameters from LLaVA and only training inserted modules. The learning rate is set to 2e-6, with a 0.03 warm-up ratio and a cosine learning rate scheduler. The batch size is 64 and no weight decay is applied. Additionally, a maximum gradient norm of 1.0 is used to stabilise convergence. Training runs on 8 A100 (80G) GPUs. During inference, we evaluate using three different token kept ratios ($\rho = 0.6$, $\rho = 0.5$, and $\rho = 0.45$) without tuning any model parameters. All inference TFLOPs reported in this paper are computed using a dummy input consisting of one image and 30 text tokens.

4.2 Dataset and Benchmarks

To prove LVPruning's data efficiency, we use a subset of the training data for LLaVA-1.5. The LLaVA-1.5 Vision Instruction Tuning dataset (Liu et al., 2023a) consists of 665k data samples. We remove all entries without image inputs, which results in approximately 620k training samples, and the model is trained for one epoch. To evaluate the performance and computational efficiency of LVPruning, we calculate its inference FLOPs and assess it on nine multi-modal benchmark datasets. These include VQAv2 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), VizWiz (Gurari et al., 2018), SciQA-IMG (Lu et al., 2022), and TextQA (Singh et al., 2019), with top-1 accuracy (acc@1) used as the evaluation metric for all these benchmarks. Additionally, POPE (Li et al., 2023b) is assessed using the F1-score across three splits. MMBench (Liu et al., 2023c) is evaluated through multiple-choice questions on both English and Chinese-translated versions. LLaVA-Wild (Liu et al., 2023b) and MM-Vet (Yu et al., 2024) assess model responses with the assistance of GPT-4. The details about each benchmark can be found in Appendix A.

5 Experimental Results

In this section, we analyse the experimental results of LVPruning across nine multi-modal benchmarks. Section 5.1 examines the performance and inference TFLOPs of LVPruning compared to the base MLLM, LLaVA-1.5 (Liu et al., 2023a), highlighting its effectiveness in reducing computational cost. Section 5.2 compares LVPruning with state-

Method	TFLOPs	VQAv2	GQA	Vizwiz	SQA-IMG	TextVQA
LLaVA-1.5-7B	8.38	78.5	62.0	50.0	69.4*	58.2
LVPruning ($\rho = 0.6$)	$3.97_{-52.6\%}$	$78.1_{-0.4}$	$61.5_{-0.5}$	$51.2_{+1.2}$	$69.0_{-0.4}$	58.2_{+0}
LVPruning ($\rho = 0.5$)	$3.18_{-62.1\%}$	$77.3_{-1.2}$	$60.7_{-1.3}$	$51.3_{+1.3}$	$68.7_{-0.7}$	$58.0_{-0.2}$
LVPruning ($\rho = 0.45$)	$2.79_{-66.7\%}$	$75.7_{-2.8}$	$59.3_{-2.7}$	$50.8_{+0.8}$	$68.6_{-0.8}$	$57.5_{-0.7}$

Table 1: Performance and inference TFLOPs comparison between LLaVA-1.5-7B (Liu et al., 2023a) and LVPruning on Visual Question Answering (VQA) benchmarks. LVPruning can significantly reduce the inference cost while maintaining marginal performance loss.

Method	POPE			MMBench		LLaVA	MM
	rad	pop	adv	en	cn	-Wild	-Vet
LLaVA-1.5-7B	87.3	86.1	84.2	64.3	58.3	65.4	31.1
LVPruning ($\rho = 0.6$)	$88.1_{+0.8}$	$86.5_{+0.4}$	$84.4_{+0.2}$	$64.0_{-0.3}$	$57.4_{-0.9}$	$67.9_{+2.5}$	$33.3_{+2.2}$
LVPruning ($\rho = 0.5$)	$87.6_{+0.3}$	$86.2_{+0.1}$	$84.1_{-0.1}$	$63.9_{-0.4}$	$56.3_{-2.0}$	$65.5_{+0.1}$	$31.6_{+0.5}$
LVPruning ($\rho = 0.45$)	$87.4_{+0.1}$	86.1_{+0}	$84.0_{-0.2}$	$63.4_{-0.9}$	$56.3_{-2.0}$	$60.7_{-4.7}$	$30.8_{-0.3}$

Table 2: Performance comparison between LLaVA-1.5-7B (Liu et al., 2023a) and LVPruning on visual instruction following benchmarks. LVPruning achieves competitive results, with improvements observed on three benchmarks.

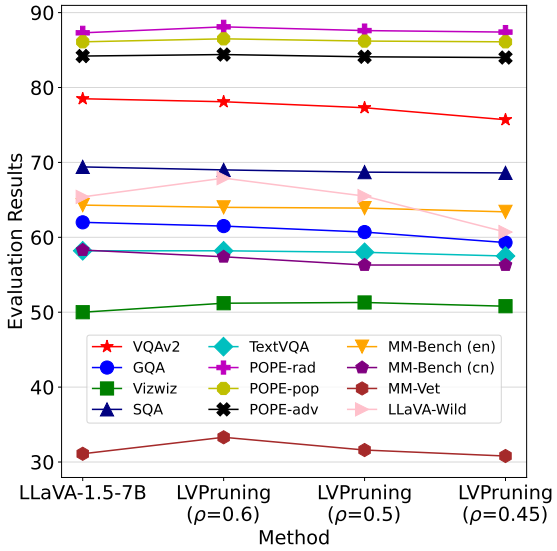

Figure 3: Performance variance between LLaVA-1.5-7B (Liu et al., 2023a) and LVPruning with different vision token kept ratios ρ on nine multi-modal benchmarks. Even at a low token kept ratio, such as $\rho = 0.45$, the performance degradation remains small.

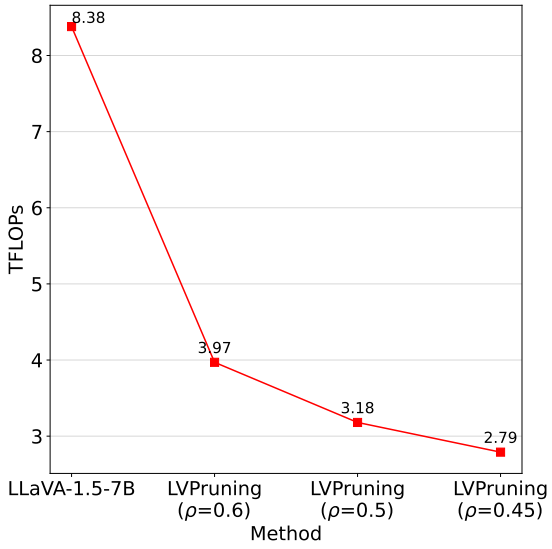

Figure 4: Comparison of Inference FLOPs of LLaVA-1.5-7B (Liu et al., 2023a) and LVPruning with different vision token kept ratio ρ.

of-the-art Q-former-based MLLMs, demonstrating its ability to balance performance and efficiency.

5.1 Performance Preservation and Computation Cost Reduction

To address research question *(i)*, we compare the model performance and inference TFLOPs of LVPruning at various vision token kept ratios, evaluating both computational savings and performance trade-off. Generally speaking, Table 1 and Table 2 show that LVPruning significantly reduces the inference cost while maintaining competitive performance. As shown in Table 1, on Visual Question Answering (VQA) benchmarks, with a pruning ratio of $\rho = 0.6$, LVPruning achieves a 52.6% reduction in TFLOPs, dropping from 8.38 to 3.97 TFLOPs, with only a minor performance

Method	TFLOPs	VQAv2	GQA	Vizwiz	SQA-IMG	TextVQA
BLIP2-14B	2.14	65.0	41.0	19.6	61	42.5
InstructBLIP-8B	1.36	–	49.2	34.5	60.5	50.1
InstructBLIP-14B	2.14	–	49.5	33.4	63.1	50.7
IDEFICS-9B	0.87	50.9	38.4	35.5	–	25.9
Qwen-VL	1.75	**78.8**	59.3	35.2	67.1	**63.8**
Qwen-VL-Chat	1.75	78.2	57.5	38.9	68.2	61.5
CrossGET	$4.39_{-47.6\%}$	77.3	**61.4**	47.7	66.7	54.9
LVPruning ($\rho = 0.5$ ours)	$3.18_{-62.1\%}$	77.3	60.7	**51.34**	**68.7**	58.0

Table 3: Performance comparison among CrossGET (Shi et al., 2023), LVPruning (ρ=0.5), and other Q-former-based MLLMs on five popular VQA benchmarks. While CrossGET attains strong results on GQA (Hudson and Manning, 2019)—it requires higher computational cost. In contrast, LVPruning achieves top performance on Vizwiz(Gurari et al., 2018)and SQA-IMG(Singh et al., 2019)) at lower TFLOPs, showcasing a favorable balance of accuracy and efficiency.

Method	POPE			MMBench		LLaVA	MM
	rad	pop	adv	en	cn	-Wild	-Vet
BLIP2-14B	**89.6**	85.5	80.9	-	-	38.1	22.4
InstructBLIP-8B	-	-	-	36.0	23.7	60.9	26.2
InstructBLIP-14B	87.7	77	72	-	-	58.2	25.6
IDEFICS-9B	-	-	-	48.2	25.2	-	-
Qwen-VL	-	-	-	38.2	7.4	-	-
Qwen-VL-Chat	-	-	-	60.6	56.7	-	-
CrossGET	84.8	84.1	82.9	**64.7**	55.2	-	-
LVPruning ($\rho = 0.5$ ours)	87.6	**86.2**	**84.1**	63.9	**56.3**	**65.5**	**31.6**

Table 4: Performance comparison among CrossGET (Shi et al., 2023), LVPruning (ρ=0.5), and other Q-former-based MLLMs on visual instruction following benchmarks. LVPruning outperforms most baselines, with only minor drops against BLIP2-14B (Li et al., 2023a) in POPE(rad) (Li et al., 2023b) and CrossGET in MMBench(en) (Liu et al., 2023c).

degradation. For instance, on the VQAv2, GQA and VizWiz benchmarks, LVPruning maintains similar accuracy, with a minimal decrease of 0.4, 0.5 and even an increase of 1.2 points, respectively. Even with higher pruning ratios, such as $\rho = 0.45$ (66.7% TFLOPs reduction), the model's performance on certain benchmarks like VizWiz (+0.8) and SQA-IMG (-0.8) are still comparable to LLaVA-1.5-7B. Table 2 shows the results for visual instruction following benchmarks. On the POPE and MMBench (en and cn) benchmarks, LVPruning yields similar or improved performance, with performance gains of up to 0.8 points on POPE. Notably, the LLaVA-Wild and MM-Vet benchmarks show obvious performance gains with $\rho = 0.6$ which increases by 2.5 and 2.2 points, respectively.

Overall, Figure 3 visualises the performance variance between LLaVA-1.5 (Liu et al., 2023a) and LVPruning with different vision token kept ratios ρ on nine multi-modal benchmarks, and Figure 4 demonstrates their inference FLOPs. LVPruning demonstrates that significant computational savings can be achieved with minimal impact on performance. Even at a high pruning ratio, such as $\rho = 0.45$, where TFLOPs are reduced by 66.7%, the performance degradation remains relatively small, indicating that LVPruning effectively balances model efficiency with task performance. Notably, the extra computation cost introduced by the inserted decision modules is 0.71 TFLOPs.

5.2 Comparisons with state-of-the-art MLLMs

To address research question *(ii)*, we use LVPruning with $\rho = 0.5$ as a representative configuration to compare against state-of-the-art Q-former-based

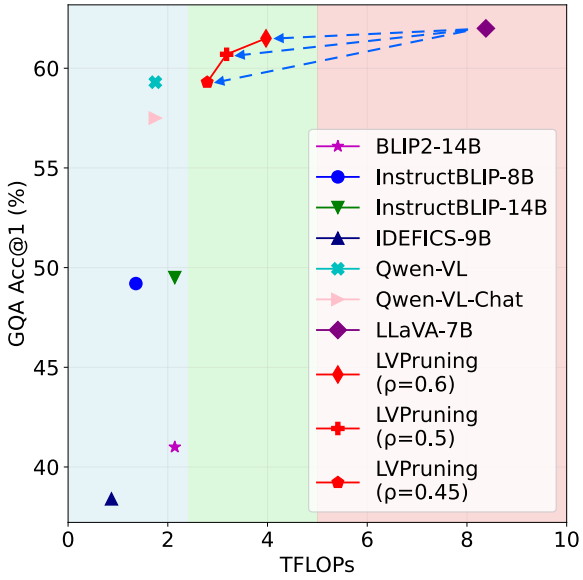

Figure 5: The relationship between inference TFLOPs and the performance of LVPruning and other state-of-the-art MLLMs evaluated on the GQA benchmark (Hudson and Manning, 2019). LVPruning (green area) outperforms Q-former-based models (blue area) while achieving substantial computational savings compared to LLaVA-1.5 (red area) (Liu et al., 2023a). This demonstrates LVPruning's ability to balance performance and efficiency among state-of-the-art MLLMs.

MLLMs. We further compare LVPruning with CrossGET, which is a token ensemble method for accelerating MLLMs (Shi et al., 2023). Generally speaking, Tables 3 and 4 demonstrate that LVPruning achieves superior performance with competitive inference FLOPs.

As shown in Table 3, on VQA benchmarks, LVPruning with $\rho = 0.5$ outperforms several state-of-the-art models, such as BLIP2-14B (Li et al., 2023a), InstructBLIP-14B (Dai et al., 2023), and IDEFICS-9B (Laurençon et al., 2023). For example, LVPruning achieves a higher VQAv2 accuracy (77.3) compared to BLIP2-14B (65.0) and IDEFICS-9B (50.9), while utilising only 3.18 TFLOPs, which is relatively higher than IDEFICS-9B but remains efficient compared to other models like BLIP2 and Qwen-VL (Bai et al., 2023). In tasks such as VizWiz, LVPruning also achieves a notable boost in performance (51.34), surpassing BLIP2 and InstructBLIP models by a large margin. Table 4 shows that on visual instruction following benchmarks, LVPruning consistently delivers competitive results. For instance, LVPruning achieves 87.6 accuracy on the POPE (rad) benchmark, closely trailing BLIP2-14B's 89.6. However, it outperforms BLIP2 on multiple datasets, such as MMBench (en) and LLaVA-Wild, with scores of 63.9 and 65.5, respectively. These results illustrate that LVPruning maintains competitive performance across various instruction following benchmarks, balancing between efficiency and effectiveness. Across both Table 3 and Table 4, CrossGET (Shi et al., 2023) and LVPruning achieve competitive results while trading off performance against efficiency. CrossGET excels on GQA and MMBench (en) but incurs higher TFLOPs. By pruning half the visual tokens (ρ=0.5), LVPruning delivers strong or superior results on multiple tasks at reduced computational cost. Figure 5 further illustrates the relationship between inference TFLOPs and the performance of LVPruning and state-of-the-art MLLMs on the GQA benchmark. LVPruning, represented by red points, showcases a balanced trade-off between computational efficiency and performance.

6 Conclusion

In this work, we introduce LVPruning, a novel language-guided vision token pruning method that can be integrated into existing MLLMs with minimal architectural changes. LVPruning computes relevance scores for each vision token based on language tokens, progressively removing redundant tokens throughout the LLM. With its middle layer, it eliminates up to 90% of vision tokens, achieving a 62.1% reduction in FLOPs with only a $\sim$0.45% average performance loss across nine multi-modal benchmarks. This makes LVPruning a practical solution for enhancing MLLM efficiency while preserving performance in multi-modal tasks.

7 Limitations

While LVPruning shows significant promise in reducing computational load, several limitations to our study should be acknowledged. Our evaluation has been conducted on a specific set of benchmarks. The performance of LVPruningin real-world applications remains unexplored. Therefore, although LVPruning is effective in reducing computational overhead, it is crucial to consider the specific requirements when applying this method to ensure that essential visual information is not compromised. Future research could involve evaluating the performance of LVPruning with human feedback to better understand its practical implications.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding...

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.

Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. 2023. Diffrate : Differentiable compression rate for efficient vision transformers. In *ICCV*, pages 17118–17128. IEEE.

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*.

Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *CoRR*, abs/1412.6115.

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 *CVPR*, pages 6325–6334. IEEE Computer Society.

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 *CVPR*, pages 3608–3617. Computer Vision Foundation / IEEE Computer Society.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

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.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. 2023. OBELICS: an open web-scale filtered dataset of interleaved image-text documents. In *NeurIPS*.

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.

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.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *CoRR*, abs/2310.03744.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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. 2023c. Mmbench: Is your multi-modal model an all-around player? *CoRR*, abs/2307.06281.

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*.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. 2023. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *CoRR*, abs/2305.17455.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Anne Treisman. 1988. Features and objects: The fourteenth bartlett memorial lecture. *The Quarterly Journal of Experimental Psychology Section A*, 40(2):201–237. PMID: 3406448.

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.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. 2019. HAQ: hardware-aware automated quantization with mixed precision. In *CVPR*, pages 8612–8620. Computer Vision Foundation / IEEE.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*.

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 *ICML*. OpenReview.net.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*. OpenReview.net.

A Benchmark Dataset Details

VQAv2 (Goyal et al., 2017) includes approximately 11k test samples and focuses on visual question answering, where models must answer questions based on images depicting various real-world scenes. GQA (Hudson and Manning, 2019), with around 12k test samples, emphasises compositional reasoning through graph-structured annotations, assessing a model's ability to understand object relationships. VisWiz (Gurari et al., 2018), containing 8k test samples, presents accessibility challenges with real-world images from visually impaired users, which are often of low quality and ambiguous, demanding robust model interpretation. SciQA-IMG (Lu et al., 2022) consists of around 4k test samples, targeting science-related visual question answering in specific domains. TextVQA (Singh et al., 2019), with 5k test samples, focuses on understanding and answering questions from textual images. POPE (Li et al., 2023b) contains approximately 9k test samples on three subsets: random, common and adversarial. It evaluates a model's ability to predict human preference judgements on hallucination of multimodal tasks. MM-Bench (en) (Liu et al., 2023c), with around 4k test samples, serves as a comprehensive benchmark for evaluating general-purpose multimodal models across various tasks, while MMBench (cn) (Liu et al., 2023c) is its Chinese translation. LLaVA-Wild (Liu et al., 2023b) has 60 test samples and emphasises answering questions about complex, in-the-wild images. Finally, MM-Vet (Yu et al., 2024) includes 218 samples and is designed to test multimodal capabilities across multiple visual and language tasks, providing a robust evaluation framework for emerging multimodal systems.