VFlowOpt: A Token Pruning Framework for LMMs with Visual Information Flow-Guided Optimization

Sihan Yang[1] Runsen Xu[1,2] Chenhang Cui[3] Dahua Lin[1,2] Tai Wang[1†] Jiangmiao Pang[1†]
[1]Shanghai AI Laboratory [2]The Chinese University of Hong Kong [3]National University of Singapore
† Corresponding Author

Abstract

Large Multimodal Models (LMMs) excel in visual-language tasks by leveraging numerous visual tokens for fine-grained visual information, but this token redundancy results in significant computational costs. Previous research aimed at reducing visual tokens during inference typically leverages importance maps derived from attention scores among vision-only tokens or vision-language tokens to prune tokens across one or multiple pruning stages. Despite this progress, pruning frameworks and strategies remain simplistic and insufficiently explored, often resulting in substantial performance degradation. In this paper, we propose VFlowOpt, a token pruning framework that introduces an importance map derivation process and a progressive pruning module with a recycling mechanism. The hyperparameters of its pruning strategy are further optimized by a visual information flow-guided method. Specifically, we compute an importance map for image tokens based on their attention-derived context relevance and patch-level information entropy. We then decide which tokens to retain or prune and aggregate the pruned ones as recycled tokens to avoid potential information loss. Finally, we apply a visual information flow-guided method that regards the last token in the LMM as the most representative signal of text-visual interactions. This method minimizes the discrepancy between token representations in LMMs with and without pruning, thereby enabling superior pruning strategies tailored to different LMMs. Experiments demonstrate that VFlowOpt can prune 90% of visual tokens while maintaining comparable performance, leading to an 89% reduction in KV-Cache memory and 3.8× faster inference.

1. Introduction

Large Multimodal Models (LMMs) [3, 8, 22, 23] have attained remarkable performance in tasks such as visual question answering [2, 13] and multimodal reasoning [45, 46], making them indispensable for applications like autonomous driving [9, 32, 36] and robotics [19, 21, 25]. To capture fine-grained visual details, modern LMMs treat images as token sequences, with models such as LLaVA-1.5 [22] processing hundreds of tokens and LLaVA-OneVision [18] handling up to several thousand tokens. Nonetheless, the increasing quantity of visual tokens significantly magnifies computational cost, memory usage, and inference time, creating a critical bottleneck for LMM deployment, particularly in resource-constrained environments or latency-sensitive scenarios.

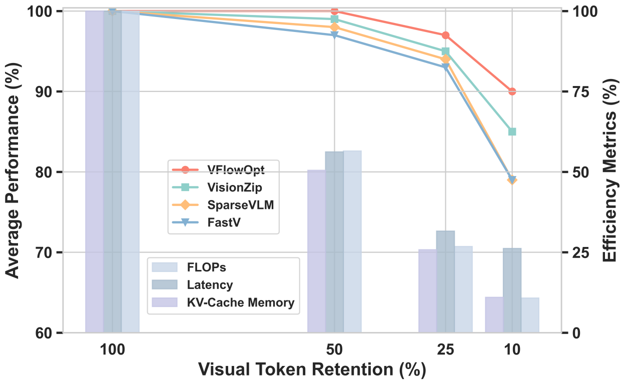

Figure 1. **VFlowOpt Performance and Efficiency.** Our VFlowOpt significantly outperforms other LMM token reduction methods across three general benchmarks (MME, MMStar, and MMBench) on LLaVA-OneVision-7B. It achieves zero performance degradation when reducing 50% of visual tokens. Furthermore, when retaining only 10% of the tokens, it achieves 90% of the original performance while reducing KV-Cache memory usage by 89% and accelerating inference by 3.8×.

To alleviate these constraints, recent work explores various methods for reducing the number of visual tokens during inference. Most existing approaches evaluate the importance of visual tokens in LMMs using attention-based importance maps derived either from vision-only tokens [41, 50] or from text-visual interactions [37, 53]. Following a predefined pruning ratio, visual tokens are pruned in a single stage [34] or progressively across multiple stages [43], often with heuristic strategies applied uniformly across different LMMs. Although promising, these methods are simplistic and underexplored, with coarse-grained pruning often causing significant performance drops due to lacking fitness for different LMMs models' characteristics.

In response to these limitations, we propose a novel token pruning framework, VFlowOpt, that devises strategies for importance map derivation and tokens selection with a visual information flow-guided method to further optimize it. Specifically, the pruning framework consists of importance map computation and a progressive pruning module with a recycling mechanism. We decompose the importance map of image tokens into two aspects: the importance of tokens in the visual context, reflected by their attention maps, and the richness of visual information in each image patch, captured by its information entropy. Based on the weighted summation of these two factors, we determine which tokens to retain or prune across multiple pruning stages and introduce a recycling mechanism to aggregate pruned tokens to avoid potential information loss. Built with this pruning strategy with several hyperparameters, a key ingredient is the visual information flow-guided method to optimize the pruning strategy, which treats the last token in each pruning stage as the most representative signal of text-visual interactions during inference. By minimizing the difference between token representations in LMMs with and without pruning, VFlowOpt ultimately delivers superior pruning strategies specifically tailored to different LMMs.

Comprehensive experiments on multiple vision-language benchmarks validate the efficacy of VFlowOpt. In particular, it substantially lowers computational cost while preserving competitive performance. As shown in Fig. 1, VFlowOpt can prune 50% of visual tokens with negligible performance loss. Moreover, it can prune 90% of visual tokens while maintaining 90% of the model's performance, resulting in an 89% reduction in KV-Cache memory usage and a 3.8× speedup in inference. These experimental results highlight its effectiveness as a practical and efficient solution for deploying LMMs in resource-constrained environments.

2. Related Work

2.1. Large Multimodal Models

Large Multimodal Models (LMMs) [3, 8, 22, 23] extend the reasoning capabilities of Large Language Models (LLMs) [4, 33, 39, 40] to vision-language tasks by integrating a pre-trained vision encoder [30, 47] with a language model, linked by an alignment module such as an MLP, or a query-based network. This design transforms visual inputs into token sequences that the LLM can process, facilitating multimodal prompts for tasks like visual question answering [2, 6, 13, 45]. To enhance performance, advanced LMMs, such as LLaVA-OneVision [18] and Qwen2-VL [35], can encode higher-resolution images into more image tokens, thereby capturing more granular visual details. However, as image resolution grows, the number of visual tokens rises exponentially, leading to substantially higher computational costs. For example, LLaVA-1.5 [22] processes 336×336 images into 576 tokens, whereas LLaVA-OneVision can handle 1152×1152 images, producing 7,290 tokens. This challenge becomes even more pronounced in video-based models like LongVA [49], which must process tokens across numerous frames. Fine-grained visual tokenization boosts LMM performance but poses an inference bottleneck, driving efforts to balance performance and cost through token reduction.

2.2. Token Reduction for LMMs

Token reduction has become a key strategy for improving the efficiency of LMMs by mitigating the computational cost associated with extensive visual token sequences. Existing methods can be broadly classified into training-based and training-free approaches. Training-based methods, such as LLaVA-Mini [51] and LLaVolta [5], introduce additional modules during model training to compress visual tokens and preserve critical information, while approaches like ATP-LLaVA [44] and p-MoD [48] train pruning modules to dynamically retain important tokens across LLM layers. However, these methods demand substantial computational resources to retrain the models, limiting their real-world applicability. In contrast, training-free methods prune tokens without additional training, often leveraging attention mechanisms to identify and discard redundant tokens. For instance, FastV[7] and SparseVLM exploit text–visual attention to rank token importance, whereas FasterVLM [50], VisionZip [41], and VTC-CLS [34] rely on [CLS] token attention in the vision encoder to evaluate token importance and prune redundant tokens. FitPrune [43] and PDrop [37] propose a progressive multi-stage pruning strategy in LMMs to fully utilize visual information. However, the pruning strategies in these methods often lack adaptability to different LMMs, which frequently leads to significant performance drops. In contrast, we propose a general framework that automatically devises superior pruning strategies for diverse LMMs, thereby bridging the gap between efficiency and performance.

3. Method

In this section, we introduce VFlowOpt, a framework that employs a pruning strategy to reduce redundant visual tokens during LMM inference while preserving essential visual information. This strategy involves multiple hyperparameters that our VFlowOpt framework automatically optimizes for proper configuration—an essential factor for maintaining model performance. Specifically, Section 3.1 explains how our method evaluates the importance of visual tokens, Section 3.2 presents progressive pruning and token

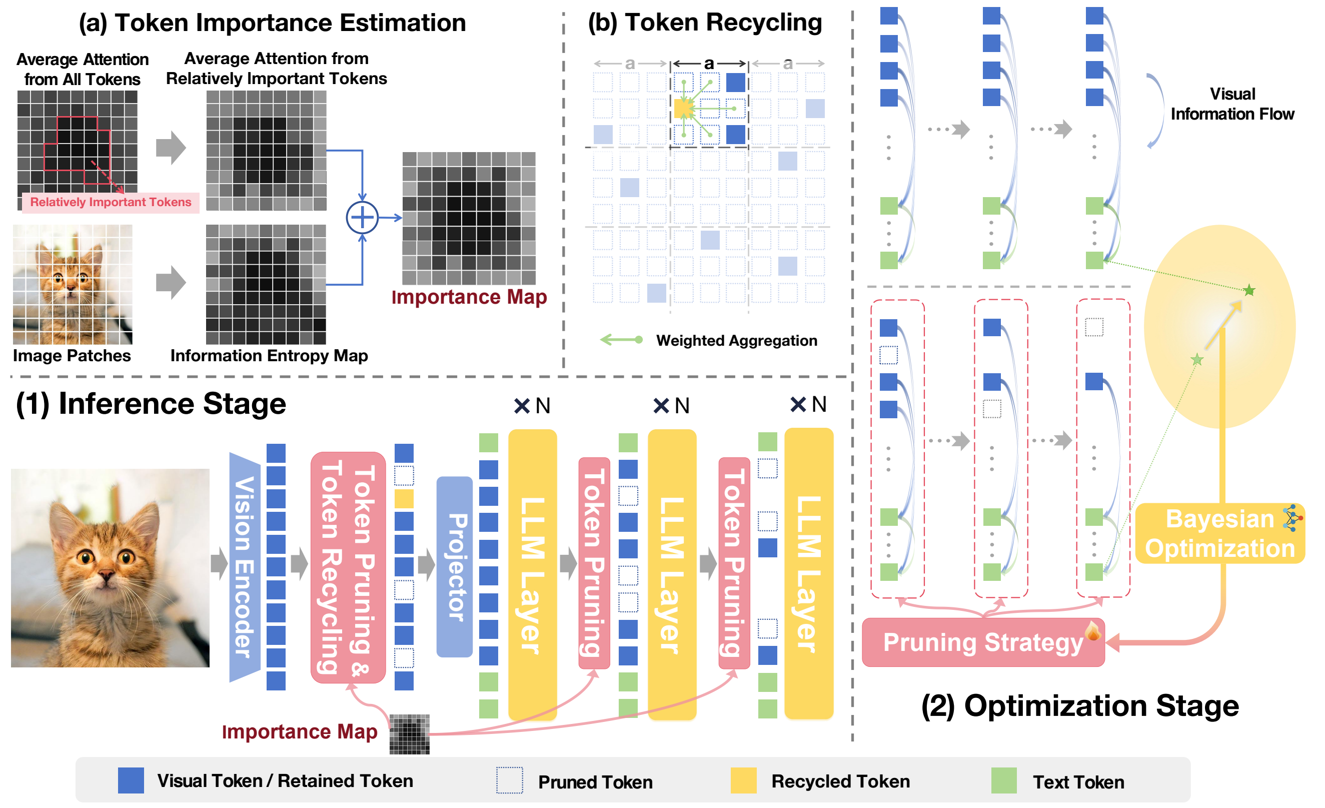

Figure 2. **Overview of VFlowOpt.** (1) During inference, VFlowOpt first assesses the importance of visual tokens, based on which progressive token pruning is performed. After the initial pruning stage, the pruned tokens are merged and recycled. The pruning strategy used in this process is defined by the (2) Optimization Stage. (a) The importance map is computed by combining the attention of relatively important tokens with the entropy of image patches. (b) The pruned tokens are grouped into grid cells, where each cell has a side length of a. Within each grid cell, the pruned tokens are fused using a weighted average, with their importance values as weights, and then recycled. (2) VFlowOpt optimizes the pruning strategy by minimizing the discrepancy of the last token in the final layer of the LMM with and without applying the pruning strategy.

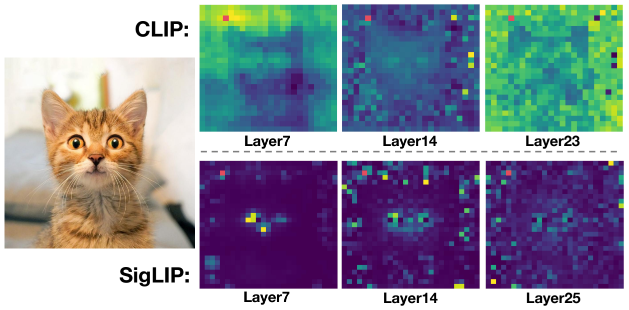

Figure 3. The attention of redundant tokens (marked in red) fails to reflect the importance of other tokens and instead focuses on similar tokens, such as background elements.

recycling strategies, and finally, Section 3.3 details how VFlowOpt customizes the superior pruning strategy (i.e., selects the pruning hyperparameters) for different LMMs.

3.1. Visual Token Importance Estimation

For a general approach to evaluate visual token importance, the previous works [15, 42] propose that the importance of visual tokens can be estimated using the average attention from all other tokens in ViTs. Although it has achieved some success, we find that redundant tokens (e.g., tokens corresponding to background regions) often assign disproportionately high attention to other similar redundant tokens (shown in Fig. 3), reducing the reliability of the importance estimation. To address this, we first identify relatively important tokens based on the attention they receive from all tokens (tokens receiving higher attention are considered relatively important), as shown in Fig. 2 (a). We then exclude redundant tokens and use the attention from these relatively important tokens as a more robust metric. Additionally, we incorporate the information entropy of image patches into the importance score to prioritize tokens corresponding to visually informative regions.

To identify relatively important visual tokens, we define a threshold:

$$\tau = t \cdot \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij}, \tag{1}$$

where τ is the threshold, A_{ij} is the attention weight from token i to token j within a ViT layer, N is the total number of tokens, and t is a sensitivity hyperparameter. We then treat tokens with a total attention exceeding τ as relatively important:

$$\mathcal{K} = \{j \mid \sum_{i=1}^{N} A_{ij} > \tau\}. \tag{2}$$

Here, $\mathcal{K}$ denotes the set of indices of these relatively important tokens.

The information entropy of an image patch corresponding to token i is defined as:

$$H(V_i) = -\sum_{k=0}^{L-1} p_k \log p_k, \tag{3}$$

where V_i is the image patch corresponding to token i, p_k is the proportion of pixels in the patch with gray level k, and L is the total number of possible gray levels (256 for 8-bit images). Here, the gray level is defined as the average of the RGB channel values, providing a scalar representation of pixel brightness. Higher entropy reflects greater diversity in pixel intensities and richer visual information.

The importance score of token i is computed by combining the attention it receives from tokens in $\mathcal{K}$ with the entropy of its corresponding image patch, normalized via the softmax function:

$$I_i = \sum_{k \in \mathcal{K}} A_{ki} \;+\; \alpha \cdot \frac{\exp\big(H(V_i)\big)}{\sum_{j=1}^{N} \exp\big(H(V_j)\big)}, \tag{4}$$

where I_i denotes the importance score of token i, A_{ki} is the attention weight from token k to token i, $H(V_i)$ is the entropy of the image patch corresponding to token i, α is a hyperparameter that determines the entropy term's contribution, and N is the total number of tokens. This approach mitigates biases from redundant tokens and leverages the visual information of image patches, thereby facilitating effective estimation of visual token importance.

3.2. Progressive Pruning and Token Recycling

Previous studies [38, 52] indicate that visual tokens play a more critical role in the shallower layers of LMMs, whereas redundancy tends to increase in deeper layers. Based on this, we adopt a progressive pruning strategy. To balance the pruning process's simplicity with its fine-grained configuration, we evenly divide the LMM into three stages. At the beginning of each stage, a predetermined fraction of visual tokens with higher importance scores (as defined in Sec. 3.1) is retained according to the stage-specific retention ratios $R = [R_1, R_2, R_3]$, while the rest are pruned. Notably, the position IDs of visual tokens remain unchanged following pruning, preserving the original spatial structure of the visual input. Following the previous work [38], we compute the average visual token retention rate across the entire LMM as follows:

$$\overline{R} = \frac{R_1 \cdot L_1 + R_1 \cdot R_2 \cdot L_2 + R_1 \cdot R_2 \cdot R_3 \cdot L_3}{L}, \tag{5}$$

where L_1, L_2, and L_3 represent the number of layers in the three stages, respectively, and L denotes the total number of layers in the LMM.

To prevent the loss of any small but potentially significant information during the initial pruning process (before visual tokens are fed into the LLM), we propose a token merging and recycling strategy to compactly represent redundant visual information. Specifically, We use a square grid (with each cell having side length a) to group pruned tokens that fall into the same grid cell, as shown in Fig. 2 (b). Within each cell, the pruned tokens are fused into a single token by computing a weighted average of their representations, using their importance scores as weights. This fused token then replaces the pruned token with the highest importance score in that cell and is incorporated into the set of retained tokens.

Formally, a token t_i with spatial coordinates (x_i, y_i) belongs to a grid cell $\mathcal{G}_{p,q}$ if

$$\lfloor x_i/a \rfloor = p \quad \text{and} \quad \lfloor y_i/a \rfloor = q, \tag{6}$$

where p and q are the row and column indices of the grid cell, respectively. Suppose there are k pruned tokens in $\mathcal{G}_{p,q}$ with corresponding importance scores $I_1, I_2, \ldots, I_k$ and token representations $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k$. The fused token $\mathbf{t}^{p,q}_{\text{merged}}$ is computed as:

$$\mathbf{t}^{p,q}_{\text{merged}} = \frac{\sum_{i=1}^{k} I_i \cdot \mathbf{t}_i}{\sum_{i=1}^{k} I_i}. \tag{7}$$

We then assign $\mathbf{t}^{p,q}_{\text{merged}}$ to the position of the pruned token with the highest importance score I_{max} in $\mathcal{G}_{p,q}$, counting it among the retained tokens. By combining token pruning with this recycling strategy, we reduce token redundancy while avoiding potential information loss.

3.3. Pruning Strategy Optimization

The pruning strategies described in Sec. 3.1 and Sec. 3.2 involve several key hyperparameters that directly affect the performance of LMMs after token pruning. Properly defining these hyperparameters is crucial for preserving model performance. However, existing approaches often rely on manually designed pruning strategies and apply the same strategy across different LMMs, without considering the unique characteristics of each model. This coarse-grained approach can lead to significant performance degradation due to its lack of fitness for different LMMs, leaving the task of designing pruning strategies tailored to specific LMMs as a formidable challenge.

Previous interpretability studies of LMMs [16, 54] offer crucial insights into their internal mechanisms. Specifically, these works reveal that in the lower and middle layers of LMMs, visual information from visual tokens is aggregated into the corresponding query text tokens. In the higher layers, the multimodal representation encoded in the query text tokens is further progressively propagated to the final position of the input sequence, ultimately influencing the subsequent inference process.

Inspired by this insight, we propose a framework that requires only a small amount of unlabeled data and leverages the internal flow of visual information within LMMs to search for the superior pruning strategy tailored to the characteristics of different LMMs, as shown in Fig. 2 (2). Specifically, we recast the task of designing the superior pruning strategy as an optimization problem, aiming to minimize the discrepancy in visual information flow with and without applying a pruning strategy to perform visual token pruning. In this framework, we treat the final token in the last layer as the representative outcome of the visual information flow. We use cosine similarity to measure the similarity of the final token representation with and without visual token pruning. A higher similarity indicates that the discrepancy between the visual information flow with and without pruning is smaller. Accordingly, we define the following optimization objective:

$$\max_{s \in \mathcal{S}} f(s) = \text{CosineSim}(h_f, g_s(h_f)). \tag{8}$$

Here, h_f represents the representation of the final token in the last layer before pruning, and $g_s(h_f)$ represents the final token representation after applying the pruning strategy s (with g_s modeling its effect on the token feature), and $\mathcal{S}$ is the solution space.

To efficiently search for the superior pruning strategy s, we employ Bayesian optimization, which systematically explores the hyperparameter space—including the threshold sensitivity t, the entropy weight α, the grid size a, and the pruning ratios R_1, R_2, and R_3—to maximize the target function $f(s)$. In short, Bayesian optimization constructs a surrogate model to approximate the objective function and employs an acquisition function to balance exploration and exploitation, thereby efficiently guiding the search for promising hyperparameter settings. The details of this optimization process are presented in Algorithm 1.

4. Experiments

In this section, we evaluate our approach on various LMMs across diverse image and video benchmarks, followed by an efficiency analysis and an ablation study of each component. Finally, we illustrate how our method affects discrepancies in the visual information flow with and without pruning, offering deeper insights into our approach.

Algorithm 1 VFlowOpt with Bayesian Optimization

```
Input: LMM $\theta$, data samples $\mathcal{D}$, number of Bayesian optimization iterations $T$, computation budget $\overline{R}$, target function $f(\cdot)$: the sum of cosine similarities between the last token in the last layer of the LMM $\theta$ with and without visual token pruning, computed over all data samples $\mathcal{D}$.
Output: retention rates $R_1$, $R_2$, $R_3$; threshold sensitivity $t$; entropy weight in importance score $\alpha$; grid size $a$.
Initialize a Gaussian Process model $\mathcal{GP}$
Define the acquisition function $A(\cdot)$ (Expected Improvement is adopted)
Uniformly sample initial points:
    $X_0 = \{(R_1, R_2, t, \alpha, a) \mid R_v, R_1, t, \alpha, a \in \text{valid ranges}\}$
for all $x \in X_0$ do
    Calculate $R_3$ using the constraint: $\overline{R} = (R_1 \cdot L_1 + R_1 \cdot R_2 \cdot L_2 + R_1 \cdot R_2 \cdot R_3 \cdot L_3)/L$
    Form pruning strategy: $s = (R_1, R_2, R_3, t, \alpha, a)$
    Evaluate the target function $f(s \mid \theta, \mathcal{D})$
end for
for $n = 0$ to $T - 1$ do
    Fit $\mathcal{GP}$ to the observed data $(S_n, f(S_n \mid \theta, \mathcal{D}))$
    Select the next point: $x_{n+1} \leftarrow \arg\max_x A(x; \mathcal{GP})$
    Calculate $R_3$ using the constraint: $\overline{R} = (R_1 \cdot L_1 + R_1 \cdot R_2 \cdot L_2 + R_1 \cdot R_2 \cdot R_3 \cdot L_3)/L$
    Form pruning strategy: $s_{n+1} = (R_1, R_2, R_3, t, \alpha, a)_{n+1}$
    Update $S_{n+1} \leftarrow S_n \cup \{s_{n+1}\}$
end for
return $(R_1, R_2, R_3, t, \alpha, a)^*$ that maximize $f$
```

4.1. Experiment Setting

Datasets. We evaluate our method on ten image-based multimodal benchmarks: GQA [1], VizWiz [14], ScienceQA-IMG [27], TextVQA [31], ChartQA [28], POPE [20], MME [10], MMBench [26], MMStar [6], and DocVQA [29]. For video understanding, we adopt two datasets—SeedBench (video) [17] and VideoMME [11], where VideoMME is partitioned by video length into short, medium, and long subsets. Further details are presented in the Appendix.

Model Architectures. We integrate VFlowOpt into multiple LMMs, including LLaVA-OneVision-7B, LLaVA-NeXT-7B [24], and Qwen2-VL-7B [35]. These models employ various vision encoders (SigLIP, CLIP, and a ViT designed for Qwen2-VL) and LLM backbones (Qwen2-7B and Vicuna-7B). To prevent out-of-memory issues in Qwen2-VL, we set *max_pixels = 3000000*.

Comparison Methods. We compare VFlowOpt against five baseline methods: FastV [7], FitPrune [43], PDrop [37], SparseVLM [53], and VisionZip [41]. FastV,

Method	MMStar	MME	MMB	SQA	POPE	GQA	VizWiz	VQA$^{\text{Text}}$	ChartQA	DocVQA	Avg.
Upper Bound, 100% Tokens											
Vanilla	61.7	1581	80.8	95.8	89.1	62.2	60.4	76.0	80.0	87.5	100%
Retain 50% Tokens (↓ 50%)											
FastV (ECCV24)	58.9	1549	79.4	92.8	87.9	61.5	**61.1**	72.5	68.6	84.0	96.5%
FitPrune (AAAI25)	4.5	96.6	1.0	8.6	18.8	16.2	6.8	37.6	37.7	30.7	21.4%
PDrop (CVPR25)	37.1	628.6	32.1	64.8	81.3	44.2	48.9	63.7	62.7	62.2	71.3%
SparseVLM (2024.10)	59.8	1577	80.5	94.1	88.1	61.9	60.4	73.9	70.5	80.8	97.1%
VisionZip (CVPR25)	60.4	1587	80.3	94.6	89.3	**62.7**	59.8	74.2	75.4	88.4	98.9%
VFlowOpt	**61.3**	**1591**	**81.1**	**95.4**	**89.4**	62.4	60.0	**75.1**	**77.8**	**90.0**	**99.9%**
Retain 25% Tokens (↓ 75%)											
FastV (ECCV24)	54.0	1539	77.0	88.6	83.8	58.2	**61.0**	58.3	42.7	62.9	86.4%
FitPrune (AAAI25)	5.2	92.6	1.1	5.9	20.2	14.5	5.1	29.7	27.3	23.2	17.6%
PDrop (CVPR25)	36.8	574.4	31.3	62.8	78.0	42.2	47.1	56.3	47.5	47.0	65.2%
SparseVLM (2024.10)	56.6	1520	78.7	90.3	87.2	59.7	60.8	66.3	54.0	66.6	90.5%
VisionZip (CVPR25)	54.6	1562	78.9	90.4	88.8	61.0	60.4	70.0	66.3	79.6	94.3%
VFlowOpt	**57.8**	**1570**	**79.9**	**92.3**	**89.1**	**61.2**	60.4	**72.5**	**69.1**	**82.3**	**96.3%**
Retain 10% Tokens (↓ 90%)											
FastV (ECCV24)	46.0	1209	70.1	81.7	77.0	51.5	56.4	35.6	21.3	33.2	69.7%
FitPrune (AAAI25)	5.4	27.0	0.9	3.5	2.3	4.6	0.7	6.9	5.9	6.9	5.1%
PDrop (CVPR25)	31.8	456.3	22.5	59.0	65.9	35.7	43.3	33.7	19.6	18.5	49.9%
SparseVLM (2024.10)	45.1	1191	71.8	83.7	80.0	54.6	56.4	39.8	37.6	39.6	74.0%
VisionZip (CVPR25)	49.5	1389	74.8	86.2	**86.1**	57.2	56.8	56.4	46.1	49.0	82.1%
VFlowOpt	**52.0**	**1464**	**75.1**	**88.4**	85.2	**57.3**	**57.3**	**60.2**	**53.6**	**56.1**	**85.5%**

Table 1. Performance comparison on LLaVA-OneVision-7B under different token retention conditions. "Avg." refers to average accuracy on 10 benchmarks. For each reduction ratio, the best average performance is shown in **bold**.

FitPrune, PDrop, and SparseVLM rely on text-visual attention in the LLM to prune visual tokens but differ as follows: FastV performs a single pruning step after the second LLM layer; FitPrune prunes tokens in each layer according to predefined, layer-specific ratios; PDrop conducts four pruning steps within the LLM, using the same retention ratio each time; and SparseVLM uses the attention weights of preselected text tokens to evaluate the importance of visual tokens. VisionZip, in contrast, determines token importance based on the [cls] token's attention; for models lacking a [cls] token, we follow VisionZip's original procedure by computing the average attention each token receives from every other token in the ViTs.

Implementation Details. For LLaVA-OneVision-7B and Qwen2-VL-7B, we perform token pruning at three distinct points: before the LLM, and after the 9th and 18th layers. For LLaVA-NeXT-7B, pruning is conducted before the LLM, and again after the 10th and 20th layers. During optimization, we sample 30 unlabeled instances from each model's training datasets; for models without publicly available training dataset, we instead use random samples from the LLaVA-OneVision training set. The optimization is performed for a total of 50 iterations. For LMMs that modify visual tokens output by the vision encoder (e.g., unpadding and interpolation in LLaVA-OneVision, and unpadding in LLaVA-NeXT), we apply corresponding transformations to the importance maps so they remain fully aligned with the final visual tokens.

4.2. Image Understanding Tasks

We evaluate the proposed VFlowOpt on image understanding benchmarks with LLaVA-OneVision-7B using various pruning ratios, and present the results in Tab. 1. Compared to other baselines, VFlowOpt consistently maintains superior accuracy across different levels of token pruning. With 50% token retention, VFlowOpt achieves 99.9% of the original performance, exceeding the second-best approach by 1.0%. This negligible performance drop underscores the method's strong potential for practical deployment. Under more extreme pruning conditions with very few retained tokens, VFlowOpt's advantage becomes more pronounced. When only 10% of the visual tokens are retained, VFlowOpt preserves 85.5% of the original performance, surpassing the second-best approach by 3.3%. This finding suggests that VFlowOpt can effectively leverage limited visual information to maintain high performance under strict computational budgets. When applied to LLaVA-OneVision, both FitPrune and PDrop disrupt the model's normal output, indicating their inability to devise effective pruning strategies for different architectures. In contrast, VFlowOpt demonstrates a clear advantage by tailoring its pruning strategy to each model's unique characteristics.

To further validate VFlowOpt's generalization capability, we evaluate it on LLaVA-NeXT-7B and Qwen2-VL-7B. As shown in Tab. 2 and Tab. 3, VFlowOpt once again delivers the top results on these LMMs. When retaining 25%

Method	MMStar	MME	MMB	SQA	POPE	GQA	Avg.
Upper Bound, 100% Tokens							
Vanilla	37.6	1519	67.4	70.1	86.5	64.2	100%
Retain 25% Tokens (↓ 75%)							
FastV	35.1	1477	65.6	67.4	83.1	60.4	95.7%
VisionZip	35.8	1501	65.4	**67.9**	86.7	61.5	97.3%
VFlowOpt	**37.0**	**1514**	**67.0**	67.7	**87.6**	**62.6**	**98.8%**
Retain 10% Tokens (↓ 90%)							
FastV	29.2	1282	61.6	63.8	71.7	55.9	85.7%
VisionZip	32.6	1378	61.5	67.1	83.5	57.0	91.6%
VFlowOpt	**35.1**	**1393**	**62.9**	**67.4**	**83.6**	**57.3**	**93.4%**

Table 2. Comparative experiments on LLaVA-NeXT-7B.

Method	MMStar	MME	MMB	SQA	POPE	GQA	Avg.
Upper Bound, 100% Tokens							
Vanilla	57.5	1680	80.3	84.7	88.4	62.2	100%
Retain 25% Tokens (↓ 75%)							
FastV	54.5	1597	76.3	79.0	81.8	57.2	93.8
VisionZip	55.2	1618	78.9	81.3	86.9	60.0	96.8
VFlowOpt	**55.9**	**1659**	**79.8**	**81.5**	**87.1**	**60.4**	**97.8**
Retain 10% Tokens (↓ 90%)							
FastV	44.9	1405	70.0	75.5	75.8	51.6	84.4
VisionZip	49.3	1518	76.5	78.2	84.3	54.1	90.9
VFlowOpt	**51.0**	**1591**	**78.0**	**78.6**	**84.5**	**54.8**	**92.8**

Table 3. Comparative experiments on Qwen2-VL-7B.

Method	SeedBench	VideoMME			Avg.
	(video)	S	M	L	
Upper Bound, 100% Tokens					
Vanilla	56.9	70.5	54.6	49.5	100%
Retain 25% Tokens (↓ 75%)					
FastV	54.7	66.9	53.2	47.7	96.2%
VisionZip	56.4	68.3	55.3	49.0	99.0%
VFlowOpt	**56.8**	**68.9**	**55.8**	**49.2**	**100%**
Retain 10% Tokens (↓ 90%)					
FastV	48.7	53.7	47.6	42.3	83.6%
VisionZip	55.0	59.7	51.8	46.3	92.4%
VFlowOpt	**55.5**	**63.3**	**52.7**	**48.6**	**95.5%**

Table 4. Comparative experiments on video understanding tasks.

of the tokens, VFlowOpt preserves 98.8% of the original performance on LLaVA-NeXT-7B and 97.8% on Qwen2-VL-7B. Even under the stringent condition of retaining only 10% of the tokens, VFlowOpt maintains 93.4% of the performance on LLaVA-NeXT-7B and 92.8% on Qwen2-VL-7B, demonstrating its strong generalizability.

4.3. Generalization to Video Tasks

We further explore VFlowOpt's generalization across different modalities by evaluating it on video benchmarks using LLaVA-OneVision-7B. As shown in Tab. 4, VFlowOpt surpasses other baselines under various token retention ratios. Notably, when only 25% of the tokens are preserved, VFlowOpt incurs virtually no performance loss on LLaVA-OneVision-7B, maintaining 100% of its original performance. Even under the more stringent condition of retaining just 10% of the tokens, VFlowOpt still retains 95.5% of the original performance, underscoring its robust generalization in the video domain.

4.4. Efficiency Analysis

We demonstrate the efficiency of VFlowOpt by conducting a comparative study on LLaVA-OneVision-7B running on a single NVIDIA A100-SXM4-80GB GPU, focusing on FLOPs, KV-Cache memory usage, and inference latency. The efficiency analysis of other baseline methods is provided in the Appendix.

We assess various token pruning ratios on the MME benchmark, measuring overall performance and average efficiency metrics across all samples. As shown in Tab. 5, VFlowOpt significantly enhances LLaVA-OneVision-7B's computational efficiency by reducing FLOPs, shrinking KV-Cache memory usage, and accelerating inference speed. Notably, at a 50% token pruning ratio, VFlowOpt achieves a 49.5% reduction in KV-Cache memory and a 1.8× speedup in inference, while boosting performance by 0.7%. At a more aggressive 75% pruning ratio, we observe a 74.2% reduction in KV-Cache memory and a 3.1× speedup, with only a 0.7% performance drop. Under extremely tight computational budgets, pruning 90% of visual tokens reduces the KV-Cache memory footprint by 89%, accelerates inference by 3.8×, and degrades performance by only 7.4%.

Such reductions in inference latency substantially benefit user-facing applications demanding real-time performance, such as autonomous driving and robotics. Moreover, by significantly shrinking the KV-Cache, large-batch inference on LMMs can accommodate more user requests simultaneously, thereby substantially reducing overall inference costs. In summary, VFlowOpt uses far less GPU memory and delivers faster inference while preserving model performance, offering a highly practical solution for efficiently deploying LMMs in real-world scenarios.

4.5. Ablation Study

To confirm the contributions of each component in our token pruning strategy, we conduct ablation experiments on LLaVA-OneVision-7B under a computational budget of retaining 25% of tokens. As shown in Tab. 6, removing importance calibration (i.e., directly using the mean attention over all tokens received by each visual token as its importance score), omitting token recycling, or discarding progressive pruning (i.e., maintaining the same number of tokens at each layer) leads to noticeable performance degradation across MMStar, MMBench, and SQA. By contrast, the complete VFlowOpt method consistently exhibits minimal performance loss, highlighting the effectiveness of all its components in pruning redundant tokens while preserving essential visual information.

We further examine how the number of samples and the

Methods	Token Reduction	FLOPs ↓ (T)	Δ	Latency ↓ (ms)	Δ	KV Cache ↓ (MB)	Δ	Performance ↑	Δ
LLaVA-OneVision-7B	-	71.4	-	1040.1	-	1786.4	-	1581	-
+ VFlowOpt	50%	37.2	**-48.0%**	584.2	**-43.8%**	902.8	**-49.5%**	1591	+0.6%
	75%	19.1	**-73.2%**	328.5	**-68.4%**	460.6	**-74.2%**	1570	-0.7%
	90%	7.7	**-89.2%**	272.1	**-73.8%**	197.1	**-89.0%**	1464	-7.4%

Table 5. Efficiency analysis of LLaVA-OneVision-7B with VFlowOpt. The detailed metric includes computation (FLOPs), latency, and KV-Cache memory. (Δ) denotes the reduction ratio. .

Pruning Strategy	MMStar	MMBench	SQA
VFlowOpt	**57.8**	**79.9**	**92.3**
w/o Importance Calibration	56.2	79.4	91.8
w/o Token Merging	57.6	79.8	91.9
w/o Progressive Pruning	56.0	79.1	91.0

Table 6. Ablation studies of pruning strategy components.

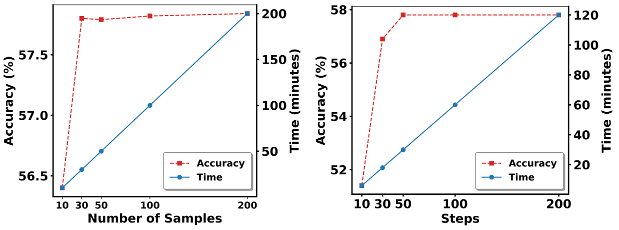

Figure 4. Relationships of sample size (left) and optimization steps (right) with optimization time and final performance.

number of optimization iterations in the Bayesian optimization procedure influence both optimization time and final performance. As illustrated in Fig. 4, the optimization time increases linearly with the number of samples and steps. Using 30 samples and 50 steps strikes an effective balance between time efficiency and performance, requiring only about 30 minutes to reach an exceptional result.

4.6. Visual Information Flow Analysis

We further investigate how Bayesian optimization formulates the superior pruning strategy by examining its impact on visual information flow. Specifically, we compare the randomly initialized pruning strategy derived from the Bayesian optimization process with its final optimized strategy, measuring their respective effects on the text tokens that follow the visual tokens at each LLM layer. Because these text tokens can receive information from the visual tokens via the LLM's unidirectional attention mechanism, examining differences in these tokens with and without pruning effectively reveals how visual information flow is altered. As shown in Fig. 5, under the same computational budget, the initial strategy causes a pronounced discrepancy between text token representations with and without pruning, indicating that it significantly disrupts the visual information flow. In contrast, when using the optimized strategy, the text tokens remain much more similar to those observed without visual token pruning. This important finding suggests that the optimization process yields a pruning strategy that preserves crucial visual information by minimizing the difference in visual information flow with and without pruning, thereby retaining LMM performance to the greatest extent possible.

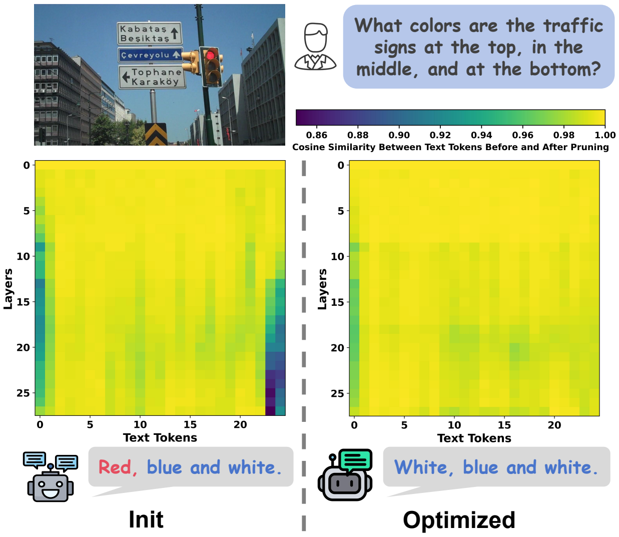

Figure 5. Visualization of the discrepancies in visual information flow corresponding to the pruning strategies before and after optimization, along with an example of visual question answering.

5. Conclusion

In this paper, we address the pressing challenge of computational inefficiency in LMMs by introducing a general framework for visual token pruning. We reformulate pruning as an optimization problem focused on minimizing the divergence of visual information flow with and without pruning, thereby facilitating a tailored pruning strategy for diverse LMMs. By leveraging calibrated attention in ViTs to evaluate token significance and merging pruned tokens using importance-based weighting, we preserve critical information during inference without requiring retraining or manual tuning. Extensive experiments underscore the efficiency and generalizability of our approach, making it a practical solution for deploying LMMs in real-world settings, particularly in resource-constrained environments.

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023. 5, 1

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2

[4] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024. 2

[5] Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Llavolta: Efficient multi-modal models via stage-wise visual context compression. *arXiv preprint arXiv:2406.20092*, 2024. 2

[6] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 2, 5, 1

[7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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, 2024. 2, 5, 1

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pages 24185–24198, 2024. 1, 2

[9] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, Tianren Gao, Erlong Li, Kun Tang, Zhipeng Cao, Tong Zhou, Ao Liu, Xinrui Yan, Shuqi Mei, Jianguo Cao, Ziran Wang, and Chao Zheng. A survey on multimodal large language models for autonomous driving, 2023. 1

[10] 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. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024. 5, 1

[11] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 5, 2

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 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)*, 2017. 1

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2017. 1, 2

[14] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 5, 1

[15] Lianyu Hu, Fanhua Shang, Liang Wan, and Wei Feng. illava: An image is worth fewer than 1/3 input tokens in large multimodal models. *arXiv preprint arXiv:2412.06263*, 2024. 3

[16] Omri Kaduri, Shai Bagon, and Tali Dekel. What's in the image? a deep-dive into the vision of vision language models. *arXiv preprint arXiv:2411.17491*, 2024. 5

[17] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 5, 2

[18] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. 1, 2

[19] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation, 2023. 1

[20] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 5, 1

[21] Yuelei Li, Ge Yan, Annabella Macaluso, Mazeyu Ji, Xueyan Zou, and Xiaolong Wang. Integrating lmm planners and 3d

skill policies for generalizable manipulation. *arXiv preprint arXiv:2501.18733*, 2025. 1

[22] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2

[24] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 5

[25] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation, 2024. 1

[26] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 5, 1

[27] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 5, 1

[28] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 5, 1

[29] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 5, 1

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[31] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 5

[32] Haoxiang Tian, Xingshuo Han, Guoquan Wu, Yuan Zhou, Shuo Li, Jun Wei, Dan Ye, Wei Wang, and Tianwei Zhang. An llm-enhanced multi-objective evolutionary search for autonomous driving test scenario generation. *arXiv preprint arXiv:2406.10857*, 2024. 1

[33] 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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 2

[34] Ao Wang, Fengyuan Sun, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. [cls] token tells everything needed for training-free efficient mllms. *arXiv preprint arXiv:2412.05819*, 2024. 1, 2

[35] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 5

[36] Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, Hao Tian, Lewei Lu, Xizhou Zhu, Xiaogang Wang, Yu Qiao, and Jifeng Dai. Drivemlm: Aligning multimodal large language models with behavioral planning states for autonomous driving, 2023. 1

[37] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024. 1, 2, 5

[38] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024. 4

[39] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 2

[40] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin,

Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 2

[41] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024. 1, 2, 5

[42] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*, 2024. 3

[43] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. *arXiv preprint arXiv:2409.10197*, 2024. 1, 2, 5

[44] Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. *arXiv preprint arXiv:2412.00447*, 2024. 2

[45] 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, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*, 2024. 1, 2

[46] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2024. 1

[47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 2

[48] Jun Zhang, Desen Meng, Ji Qi, Zhenpeng Huang, Tao Wu, and Limin Wang. p-mod: Building mixture-of-depths mllms via progressive ratio decay. *arXiv preprint arXiv:2412.04449*, 2024. 2

[49] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 2

[50] Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster, 2024. 1, 2

[51] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *The Thirteenth International Conference on Learning Representations*. 2

[52] Xiaofeng Zhang, Yihao Quan, Chen Shen, Xiaosong Yuan, Shaotian Yan, Liang Xie, Wenxiao Wang, Chaochen Gu, Hao Tang, and Jieping Ye. From redundancy to relevance: Information flow in lvlms across reasoning tasks. *arXiv preprint arXiv:2406.06579*, 2024. 4

[53] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024. 1, 5

[54] Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. *arXiv preprint arXiv:2411.18620*, 2024. 5

VFlowOpt: A Token Pruning Framework for LMMs with Visual Information Flow-Guided Optimization

Supplementary Material

A. Overview of Baselines

- FastV[7] is a plug-and-play method that optimizes inference efficiency in LMMs by dynamically pruning visual tokens after the second layer, significantly reducing computational costs while maintaining performance. It identifies that image tokens receive drastically lower attention in LLM and strategically removes less impactful tokens.
- FitPrune [43] is a training-free method for pruning visual tokens in multimodal LMMs, based on quickly estimating optimal pruning schemes through attention distribution fitting. It statistically determines which tokens can be discarded by minimizing divergence between attention distributions before and after pruning, using only a small batch of inference data. This approach rapidly produces a pruning recipe tailored to a given computation budget, significantly reducing computational complexity while preserving model performance.
- Pdrop [37] accelerates large vision-language models by progressively removing redundant visual tokens in deeper layers based on token similarity. It partitions models into multiple stages, maintaining all tokens initially to preserve critical visual information, then gradually pruning tokens as layers deepen. This approach effectively reduces computational costs without compromising performance during both training and inference.
- Sparsevlm [53] introduces a training-free, text-guided visual token sparsification method for LMMs, significantly reducing computational overhead by adaptively selecting important visual tokens based on relevant text prompts. It employs an adaptive pruning strategy at each layer and recycles pruned visual tokens into compact representations to minimize information loss.
- Visionzip [41] is a simple yet effective method that reduces visual token redundancy in LMMs by selecting only the most informative tokens, significantly improving efficiency while maintaining performance. It employs a text-agnostic approach that merges and compresses redundant tokens, reducing computational costs and enhancing inference speed without requiring additional training.

B. Overview of Benchmarks

- MME [10] offers a robust benchmark for evaluating LVLMs across multimodal tasks. It assesses models on two major fronts: perception and cognition, using 14 well-structured subtasks that challenge their interpretive and analytical abilities.
- MMBench [26] takes a two-pronged approach by introducing an extensive dataset that broadens the scope of evaluation questions and a novel CircularEval strategy that utilizes ChatGPT to convert free-form responses into structured answer choices.
- ScienceQA [27] focuses on evaluating multi-hop reasoning and interpretability within scientific domains. It features a large dataset of approximately 21K multiple-choice questions across a variety of science topics, accompanied by detailed annotations and explanations.
- VizWiz [14] stands out in the VQA field by using a dataset of over 31,000 visual questions that come from a real-world setting, featuring images taken by visually impaired individuals and their associated spoken queries, along with crowdsourced answers.
- GQA [1] is built for complex visual reasoning tasks, containing 22 million questions generated from scene graph-based structures. It incorporates innovative evaluation metrics focused on consistency, grounding, and plausibility, pushing the boundaries of vision-language evaluation.
- POPE [20] introduces a methodology to evaluate object hallucination in LVLMs, transforming the task into a binary classification problem. By using simple Yes-or-No prompts, POPE highlights model tendencies towards hallucination through various object sampling strategies.
- VQA [12] collects complementary images such that every question in the balanced dataset is associated with a pair of similar images that result in two different answers to the question.
- ChartQA [28] is a large-scale benchmark designed for question answering on charts, focusing on both visual and logical reasoning with 9.6K human-written and 23.1K automatically generated questions. Unlike previous datasets, it includes real-world charts and open-ended questions that require mathematical operations, overcoming limitations of template-based approaches.
- DocVQA [29] is a large-scale dataset designed for Visual Question Answering (VQA) on document images, containing 50,000 questions over 12,000+ real-world documents. Unlike previous datasets, it requires models to understand both textual content and visual layout, including tables, forms, and complex structures. Baseline evaluations show a significant gap between model and human performance, highlighting the need for improved document comprehension methods.
- MMstar [6] is a new benchmark designed to address

Methods	**Token Reduction**	**FLOPs ↓ (T)**	**Δ**	**Latency ↓ (ms)**	**Δ**	**KV Cache ↓ (MB)**	**Δ**	**Performance ↑**	**Δ**
LLaVA-OneVision-7B	-	71.4	-	1040.1	-	1786.4	-	1581	-
+ VFlowOpt	50%	37.2	**-48.0%**	584.2	**-43.8%**	902.8	**-49.5%**	1591	+0.6%
+ FastV	50%	38.1	**-46.6%**	615.1	**-41.9%**	902.8	**-49.5%**	1549	-2.0%
+ VisionZip	50%	37.7	**-47.2%**	580.7	**-44,2%**	902.8	**-49.5%**	1587	+0.1%

Table 7. Efficiency analysis of LLaVA-OneVision-7B with VFlowOpt, FastV, and VisionZip. The detailed metric includes computation (FLOPs), latency, and KV-Cache memory. (Δ) denotes the reduction ratio.

issues in evaluating Large Vision-Language Models (LVLMs), specifically unnecessary visual content and unintentional data leakage, which can mislead performance assessments. It includes 1,500 carefully selected vision-dependent samples, ensuring accurate evaluation of LVLMs' true multi-modal reasoning abilities. MMStar introduces new metrics—Multi-Modal Gain (MG) and Multi-Modal Leakage (ML)—to measure actual improvements from multi-modal training, with evaluations showing GPT-4V leading in both accuracy and multi-modal efficiency.

- SeedBench [17] is a large-scale benchmark designed to evaluate the generative comprehension capabilities of Multimodal Large Language Models (MLLMs), featuring 19K human-annotated multiple-choice questions across 12 evaluation dimensions for both images and videos. It introduces an advanced question-generation pipeline that ensures high-quality, vision-dependent evaluation samples without requiring human or GPT intervention for assessment. SEED-Bench provides comprehensive insights into the strengths and limitations of 18 leading models, offering a robust leaderboard for continuous benchmarking and future research direction in multimodal AI.
- VideoMME [11] is the first comprehensive benchmark designed to evaluate Multi-Modal Large Language Models (MLLMs) in video analysis, covering 900 manually annotated videos across six diverse domains and 30 subcategories. It introduces a full-spectrum evaluation with multi-modal inputs, including subtitles and audio, and assesses models across various temporal contexts, from short clips to hour-long videos.

C. Efficiency Analysis about Baselines

We evaluate VFlowOpt, the well-performing baseline FastV, and VisionZip on efficiency metrics under the condition of retaining 50% of the tokens. With the same token retention rate, all methods showed identical KV-Cache memory usage, while FLOPs and latency exhibited slight differences, as shown in Tab. 7.