Boosting Multimodal Large Language Models with Visual Tokens Withdrawal for Rapid Inference

Zhihang Lin[1,2], **Mingbao Lin**[3], **Luxi Lin**[1], **Rongrong Ji**[1*]
[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, 361005, P.R. China
[2]Shanghai Innovation Institute
[3]Skywork AI
lzhedu@foxmail.com, linmb001@outlook.com, lewuluu@gmail.com, rrji@xmu.edu.cn

Abstract

Multimodal large language models (MLLMs) demand considerable computations for inference due to the extensive parameters and the additional input tokens needed for visual information representation. Herein, we introduce Visual Tokens Withdrawal (VTW), a plug-and-play module to boost MLLMs for rapid inference. Our approach is inspired by two intriguing phenomena we have observed: (1) the attention sink phenomenon that is prevalent in LLMs also persists in MLLMs, suggesting that initial tokens and nearest tokens receive the majority of attention, while middle vision tokens garner minimal attention in deep layers; (2) the presence of information migration, which implies that visual information is transferred to subsequent text tokens within the first few layers of MLLMs. As per our findings, we conclude that vision tokens are unnecessary in the deep layers of MLLMs. Thus, we strategically withdraw them at a certain layer, enabling only text tokens to engage in subsequent layers. To pinpoint the ideal layer for VTW, we initially analyze a limited set of tiny datasets and choose the first layer that meets the Kullback-Leibler divergence criterion. Our VTW approach can cut computational overhead by over 40% across diverse multimodal tasks while maintaining performance.

Code — https://github.com/lzhxmu/VTW

Introduction

In recent years, major progress has been made in generative AI with the development of large language models (LLMs) (Achiam et al. 2023; Team et al. 2023; Touvron et al. 2023a). Multimodal large language models (MLLMs) (Liu et al. 2023b; Li et al. 2023a; Liu et al. 2024) combine vision encoders, like CLIP (Radford et al. 2021), to extract visual features, enhancing LLM's reasoning abilities for complex tasks like visual question answering (VQA) (Lu et al. 2022; Kembhavi et al. 2016) and visual reasoning (Yue et al. 2023; Liu et al. 2023c; Fu et al. 2023).

However, MLLMs entail high inference cost due to their billions of parameters and their computational cost quadratic increases with the length of the input sequence. Converting a high-resolution image into vision tokens further increases the inference cost (Liu et al. 2024). The high inference cost of MLLMs hinders their applicability in real-time scenarios.

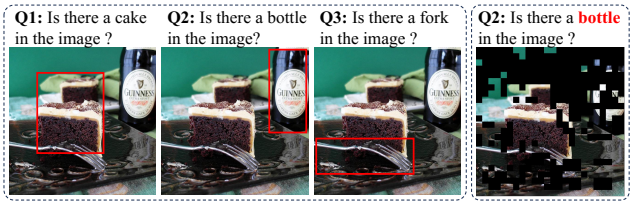

Figure 1: (a) An instance from POPE (Pope et al. 2023) with red box indicating key area for answering the question. (b) In FastV, some genuinely important tokens like "bottle" are pruned, while unimportant tokens like "cake" are preserved.

Recent methods (Shang et al. 2024; Chen et al. 2024) aim to reduce the computational cost of MLLMs through token reduction. The primary goal is to keep so-called "important" tokens based on predefined metrics and remove or merge the rest. However, these approaches have some shortcomings that need to be addressed. (1) Lack of flexibility. Methods like LLaVA-PruMerge (Shang et al. 2024) employ identical vision tokens to represent an image for different questions in VQA. However, for datasets like MME (Fu et al. 2023), POPE (Li et al. 2023b), and AI2D (Kembhavi et al. 2016), numerous questions are associated with the same image, each focusing on a distinct area of the image, as illustrated in Figure 1(a). Thus, LLaVA-PruMerge cannot dynamically adjust the important tokens based on the questions, leading to a lack of flexibility and a significant accuracy drop of 160.4 on MME benchmark. (2) Incomplete importance metric. Almost all methods necessitate the design of an importance metric for token reduction, such as cross-modal guidance for CrossGET (Shi et al. 2023) and attention score for FastV (Chen et al. 2024). However, there is limited theoretical evidence to establish which importance metric is the optimal. A low importance score for a token does not necessarily indicate that the token is unimportant. Moreover, some genuinely important tokens are pruned in these methods, as illustrated in Figure 1(b). Thus, designing a comprehensive importance metric to determine which tokens are important is challenging. (3) Incompatibility with KV Cache. KV Cache (Pope et al. 2023) is an essential approach for

*Corresponding author

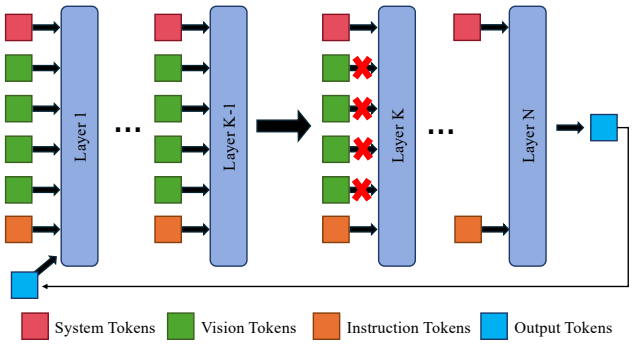

Figure 2: The framework of our method. Vision tokens are withdrawn in the K-th layer of large language models.

speeding up MLLMs decoding by storing prior tokens' Key and Value states. FastV dynamically prunes partial vision tokens after a particular layer, with retained tokens varying for each auto-regressive prediction. Thus, the cached KV of previous tokens cannot be reused because the retained vision tokens change with each prediction. To use KV Cache to speed up the decoding stage, all KV Cache of vision tokens must be maintained in the initial prediction process. However, this leads to memory occupation for pruned tokens and reduces the memory benefit of token reduction. (4) Inconsistency with Flash-attention. FastV necessitates the retention of attention score for pruning unimportant visual tokens effectively. Conversely, Flash-attention (Dao et al. 2022), despite its prevalent adoption, lacks the functionality to store the required attention scores for FastV. Since Flash-attention is used a lot, this limits how useful FastV can be. Therefore, current methods don't fully meet the practical acceleration needs of MLLMs for various complex multimodal tasks.

Driven by the above analysis, we recognize that instead of being stuck in figuring out individual solutions to address the above shortcomings, a more comprehensive approach would be to withdraw all vision tokens after a specific layer. In doing so, all vision tokens are preserved within the first few layers of MLLMs, ensuring flexibility across various multimodal tasks. Second, there is no need to design a comprehensive importance metric for selecting crucial tokens, thus avoiding the unintentional removal of important vision tokens. Third, this approach is compatible with KV Cache because all vision tokens are preserved and removed simultaneously. Finally, there is no need to store attention score for pruning unimportant vision tokens, thus this approach aligns well with Flash-attention. We conduct a thorough analysis to explore the possibility of removing all vision tokens after a specific layer. In particular, we perform extensive visualizations and uncover some intriguing phenomena: (1) The attention sink phenomenon in LLMs (Xiao et al. 2023) persists in MLLMs, where initial and nearest tokens gain the most attention, while middle vision tokens receive minimal attention. (2) As the number of generated tokens increases, vision tokens receive less attention, while text tokens garner more attention. These results are attributed to the causal self-attention operation in LLMs, which only allows tokens to attend to preceding tokens, ensuring the model's generation depends on preceding content. As a consequence of this operation, information from visual tokens migrates to subsequent text tokens through several layers of attention transformation. Therefore, the latest token tends to pay progressively less attention to vision tokens and more attention to text tokens in the deeper layers and when there are more and more generated text tokens.

In this paper, we boost MLLMs with a Visual Tokens Withdrawal (VTW) strategy for rapid inference. Given that vision tokens become less crucial in deep layers of LLMs and their information has already been absorbed by the subsequent text tokens, we propose withdrawing them in the deep layers. VTM implements a visual tokens withdrawal approach at a specific layer of MLLMs, as illustrated in Figure 2. Before this layer, computations proceed as usual; after this layer, vision tokens are removed, and only text tokens participate in the computation of deep layers. To determine the withdrawal layer, we sample a small subset of datasets. Then, we calculate the Kullback-Leibler (KL) divergence between the output logits of the standard decoding and the visual tokens withdrawal decoding in each layer. Finally, we select the first layer that meets the KL divergence criterion as the vision token withdrawal layer.

We carry out extensive experiments on various multimodal tasks, such as visual question answering (Kembhavi et al. 2016; Lu et al. 2022), hallucination evaluation (Li et al. 2023b), visual reasoning (Yue et al. 2023; Fu et al. 2023) and video understanding (Jang et al. 2017; Fu et al. 2024) to show the efficacy of our VTW. Notably, VTW can reduce over 40% FLOPs on AI2D (Kembhavi et al. 2016), SQA_image (Lu et al. 2022), MMMU_Val (Yue et al. 2023), MMB_EN (Liu et al. 2023c), POPE (Li et al. 2023b), MME (Fu et al. 2023), TGIF (Jang et al. 2017), and VideoMME (Fu et al. 2024) without compromising performance. Also, VTW is applicable to the multimodal chatbot (Liu et al. 2023b) to achieve accelerated inference with imperceptible differences in the answers.

Related Work

Multimodal Large Language Models

Large Language Models (LLMs) like GPT (Achiam et al. 2023), Gemini (Team et al. 2023), and LLaMA (Touvron et al. 2023a,b) have transformed natural language processing. They've been improved to understand not just text, but also images, video, audio, *etc*. LLaVA (Liu et al. 2023b,a) combines a CLIP visual encoder (Radford et al. 2021) with a LLaMA language decoder (Liu et al. 2023b), making it good at following instructions and understanding images. Video-LLaMA (Lin et al. 2023a) endows videos and sounds understanding, improving how it processes different types of information. However, these MLLMs use a lot of tokens to process information from different sources. For example, LLaVA (Liu et al. 2023b) uses 576 vision tokens for a 336×336 image, which escalates for higher-resolution images. This becomes a bigger issue in Video-LLaMA (Lin et al. 2023a) and LLaVA-NeXT (Liu et al. 2024). While MLLMs perform well, the high computational cost, which quadratically grows with a token number, is a big challenge.

Vision Token Reduction

Token pruning (Rao et al. 2021; Xu et al. 2022; Liang et al. 2022) and merging (Bolya et al. 2023; Marin et al. 2021) directly reduce the number of tokens, thereby decreasing the inference time and memory usage. EViT (Liang et al. 2022) and Evo-ViT (Xu et al. 2022) fuse non-critical tokens into a single token for token reduction. ToMe (Bolya et al. 2023) employs a binary soft-matching algorithm to merge redundant tokens, while Token Pooling (Marin et al. 2021) utilizes clustering for token merging. DiffRate (Chen et al. 2023b) and PPT (Wu et al. 2023) unify token pruning and merging to dynamically reduce redundant tokens. In MLLMs, CrossGET (Shi et al. 2023) and MADTP (Cao et al. 2024) introduce special tokens to align tokens of different modalities and use these special tokens to guide token reduction. Qwen-VL (Bai et al. 2023), LLaVA-UHD (Guo et al. 2024), and LLaMA-VID (Li, Wang, and Jia 2023) use query tokens via cross attention to reduce vision tokens. LLaVA-PruMerge (Shang et al. 2024) leverages the visional spatial redundancy and proposes a token reduction module that employs the similarity between the class token and spatial tokens as a key criterion for pruning and merging vision tokens. It is observed that most image tokens receive inefficient attention after the second decoder layer (Chen et al. 2024), thus half of the image tokens can be safely removed.

Methodology

Motivations

MLLMs typically comprise a pre-trained vision encoder, a cross-modal projector, and a pre-trained large language model (LLM). Herein, we utilize LLaVA (Liu et al. 2023b), a recent SOTA method, to illustrate the architecture.

The vision encoder, such as CLIP ViT-L (Radford et al. 2021), extracts visual features from an input image. These visual features are represented as a set of token sequences, referred to as vision tokens. Then, a cross-modal projector transforms these vision tokens into the text embedding space, aligning the outputs of the vision encoder with the LLM. The core is a pre-trained LLM, such as Vicuna (Chiang et al. 2023), which is tasked with understanding the multimodal context and providing an appropriate response. The integration of these components enables MLLMs to process and interpret both textual and visual data, providing a more comprehensive understanding of the inputs.

MLLMs require significant computational resources for inference, with the bulk of the computation being attributed to the LLM, given that the size of the visual encoder, such as ViT-L (0.3B), is much smaller than the LLM, such as Vicuna (7B or 13B). In MLLMs, the main computation for a decoder layer comes from multi-head attention (MHA) and feed-forward network (FFN). Assuming the input sequence length is s, the hidden embedding size is h, and the FFN up-scaling factor is 4, the computational complexity for a transformer decoder layer is (Chen et al. 2023a,b, 2024):

$$\Omega(MHA + FFN) = 2s^2h + 12sh^2, \tag{1}$$

where computational complexity is quadratically influenced by the input length s. Therefore, efficient token management is crucial for optimizing the efficiency of MLLMs.

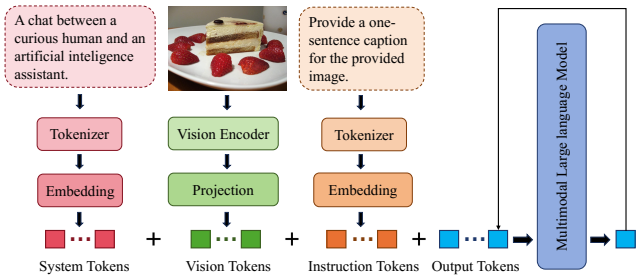

Figure 3: The illustration for the input of a multimodal large language model. The input tokens are composed of system tokens, vision tokens, instruction tokens, and output tokens.

The input tokens of LLaVA in Figure 3, consist of system tokens, vision tokens, instruction tokens, and output tokens. The system tokens are derived from fixed system prompts, which establish a dialogue system for LLaVA. Meanwhile, the instruction tokens originate from users, which specify the query question for the given image. In Figure 3, LLaVA preprocesses the input from various modalities and then concatenates all these tokens to form the inference input as:

$$X_1^t = [S_1, \mathcal{V}_1, I_1, O_1^t], \tag{2}$$

where X_i^t is inputs of the i-th layer in the t-th inference. S_1, $\mathcal{V}_1$, and I_1 denote system tokens, vision tokens, and instruction tokens, with O_1^t being the output tokens and $O_1^0 = \phi$.

The quantity of instruction tokens varies depending on user instructions and is generally fewer than the number of vision tokens. Vision tokens dominate the computation during inference by comprising the majority of input tokens. Recent works (Chen et al. 2024; Shang et al. 2024) have attempted to reduce computation by decreasing the number of vision tokens. However, these methods suffer from limited flexibility, incomplete importance metric, incompatibility with KV Cache, and inconsistency with the Flash-attention, as elaborated in the introduction.

Unnecessity of Vision Tokens in Deep Layers of MLLMs

Given that input tokens X_i^t consist of various types of tokens, it is natural to question whether the contribution of each token type to the prediction of the output token is equal or proportional to their size. Inspired by StreamingLLM (Xiao et al. 2023), we opt for attention scores as the evaluation metrics. Specifically, attention scores are derived from the causal self-attention operation within a decoder layer in LLMs. In the i-th causal self-attention layer, the hidden feature X_i^t is transformed into queries Q_i^t, keys K_i^t, and values V_i^t using three distinct learnable projection matrices. Consequently, the causal self-attention, abbreviated as CSA, can be expressed as:

$$\text{CSA}(Q_i^t, K_i^t, V_i^t) = A_i^t \cdot V_i^t, \tag{3}$$

where attention map $A_i^t = \text{Softmax}(\frac{Q_i^t {K_i^t}^T + \Lambda}{\sqrt{d}})$ and d is the hidden size of LLM. Λ is an upper triangular matrix where non-zero values are set to $-inf$ and diagonal elements are set to 0. Here, for simplicity, we omit the expression of multi-head causal self-attention. We select the last row of

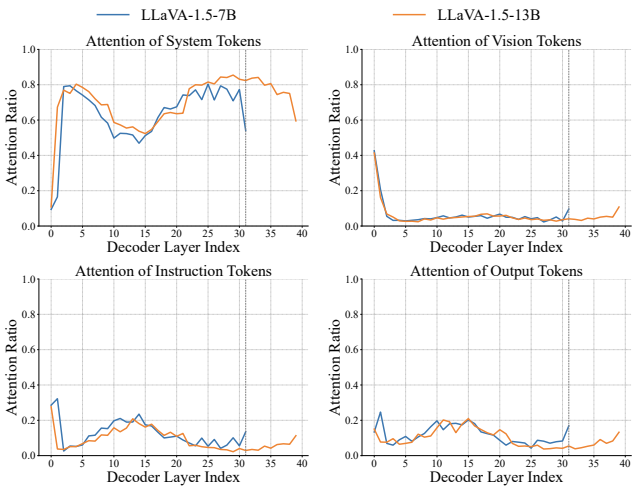

Figure 4: The output token's attention towards various input token types across different layers on a combined subset of AI2D (Kembhavi et al. 2016), MMMU_Val (Yue et al. 2023), MME (Fu et al. 2023), and POPE (Pope et al. 2023) (100 samples from each dataset). The attention values are averaged across all attention heads and output tokens.

A_i^t as the attention score α_i^t, given that only the final token is utilized to predict the output token in an auto-regressive manner. Based on the positions of system tokens, vision tokens, instruction tokens, and output tokens, we categorize α_i^t into $\alpha_i^{t,sys}$, $\alpha_i^{t,vis}$, $\alpha_i^{t,ins}$, and $\alpha_i^{t,out}$. Then, we aggregate the attention scores of each input type, resulting $\beta_i^{t,sys}$, $\beta_i^{t,vis}$, $\beta_i^{t,ins}$, and $\beta_i^{t,out}$. These variables denote the contributions of each input type to the output prediction.

We visualize β towards various input token types across different layers in Figure 4 and across different output tokens in Figure 5. Here are the main points: (1) In the initial layers, attention to system tokens increases sharply, while attention to vision tokens decreases sharply. In the middle and deeper layers, almost all the attention (80% or more) goes to just 35 system tokens, while 576 vision tokens get only 5%. (2) As more output tokens are produced, the model focuses less on vision tokens and more on these output tokens.

The "attention sink" idea from LLM (Xiao et al. 2023) helps us understand how attention changes for system tokens in different layers. In the first few layers, the output token interacts with all other tokens through causal self-attention to accumulate semantic information. Consequently, the attention directed towards system tokens is minimal due to their limited semantic content. In the deep layers, the output token possesses sufficient self-contained information for its prediction. However, owing to the nature of the softmax function, which cannot assign zero attention to undesired tokens, the output token will mostly focus on tokens with minimal semantic information, such as system tokens, to avoid incorporating undesired information from other tokens.

To explain why instruction tokens and output tokens receive greater attention than vision tokens, we introduce "information migration." Due to causal self-attention, tokens can attend only to preceding tokens. Instruction and output tokens can attend to all vision tokens, while vision tokens cannot attend to instruction tokens or output tokens. After multiple transformations of causal self-attention layers, instruction and output tokens absorb both visual and textual information, attracting more attention from output tokens.

Experimental Setting	Score
LLaVA-1.5-7B	1866.10
(a) w/o image	970.89
(b) w non-content image at 1st–16-th layers	845.39
(c) w original image at 1st–16-th layers	1872.43

Table 1: The ablation study on MME (Fu et al. 2023) under various experimental settings. In both (b) and (c), we remove vision tokens in the last 16-th layer of LLaVA-1.5-7b. "non-content " denotes a misleading image full of white area.

To verify that the information migration is completed within the first few layers of MLLMs, we design ablation studies using LLaVA-1.5-7B on MME benchmark. As demonstrated in Table 1, without any visual information, the score of LLaVA-1.5-7B decreases to 970.89, emphasizing the significance of visual information in evaluation. Interestingly, MLLMs can handle certain questions using only textual information, as evidenced by a total score of 970.89. By comparing the results of (a), (b), and (c), we observe that MLLMs achieve comparable results with baseline in the (c) setting since the correct visual information has migrated to the subsequent text tokens before 16-th layer. Besides, (b) yields a lower score than (a), as the misleading visual information has migrated to the subsequent text tokens before 16-th layer. These findings strongly support the occurrence of information migration within the initial layers of MLLMs.

Thus we can conclude that **the vision tokens are unnecessary in the deep layers of MLLMs**, as the information of vision tokens has migrated to following text tokens within the first few layers of MLLMs. Therefore, it is justifiable to withdraw all vision tokens in a specific layer of MLLMs.

Visual Tokens Withdrawal

We review the standard inference process of MLLMs, followed by a comprehensive introduction to incorporate visual tokens withdrawal into the MLLMs framework.

Remember that the symbol X_i^t represents the input of the i-th layer during the t-th inference pass of MLLMs, as in Eq. (2). The LLM employs an N-layer transformer architecture decoder to effectively predict output tokens:

$$\begin{aligned} X_{N+1}^t &= D_{1:N}(X_1^t), \\ X_1^{t+1} &= [X_1^t, P(X_{N+1}^t)], \end{aligned} \tag{4}$$

where $D_{i:j}(\cdot)$ denotes decoder layers from layer i to layer j. The $P(\cdot)$ predicts subsequent output tokens and calculates their embeddings by taking hidden features as input.

Keeping in mind that the vision tokens are not necessary in the deep layers of MLLMs due to the information migra-

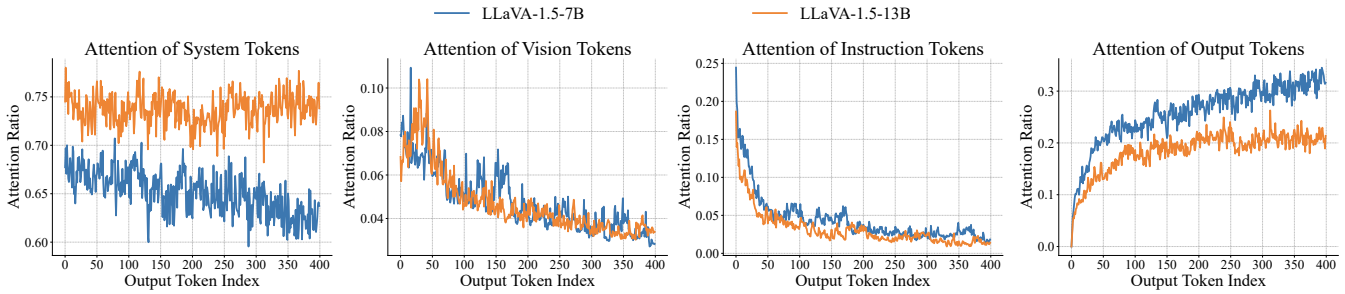

Figure 5: The output token's attention towards various input token types across output tokens. Our visualization is conducted on a subset of AI2D (Kembhavi et al. 2016), MMMU_Val (Yue et al. 2023), MME (Fu et al. 2023), and POPE (Pope et al. 2023) (20 samples from each dataset). The attention is averaged across all attention heads and layers.

tion, we proceed to withdraw these tokens at the K-th layer:

$$\begin{aligned} X_K^t &= D_{1:K-1}(X_1^t), \\ \mathcal{X}_K^t &= X_K^t - \mathcal{V}_K^t, \\ \mathcal{X}_{N+1}^t &= D_{K:N}(\mathcal{X}_K^t). \end{aligned} \tag{5}$$

In other words, before reaching the K-th layer, computations carry on as usual; However, after the K-th layer, vision tokens are withdrawn, leaving only text tokens to engage in the computation of deep layers. Figure 2 depicts an example.

Given the various variants and architectures of MLLMs, such as LLaVA-1.5-7B/13B (Liu et al. 2023b) and LLaVA-NeXT-7B (Liu et al. 2024), it is important to note that more complex tasks may require additional layers to process vision tokens for accurate predictions. Consequently, the optimal withdrawal layer K varies depending on the specific MLLMs and tasks at hand. To determine the appropriate value for K, we randomly sample a tiny subset of the target datasets for guidance. From Figure 4 that vision tokens receive minimal attention after layer 5, we enumerate K with values ranging from 5 to N and compute the KL divergence between the standard output logits and the VTW's output logits. We then select the first layer that satisfies the criterion as the withdrawal layer K:

$$KL(lm_{head}(X_{N+1}), lm_{head}(\mathcal{X}_{N+1})) < \eta, \tag{6}$$

where, $KL(\cdot)$ shows KL divergence, $lm_{head}(\cdot)$ is the LLM project head, X_{N+1} and $\mathcal{X}_{N+1}$ are calculated by Eq. (4) and Eq. (5) , and η denotes the threshold.

Experimentation

Experimental Setting

VTW serves as a seamless extension to off-the-shelf pretrained MLLMs, requiring no extra training cost. We apply VTW to popular open-source MLLMs, such as LLaVA-1.5 (Liu et al. 2023b), LLaVA-NeXT (Liu et al. 2024), and Video-LLaVA (Lin et al. 2023b). We conduct comprehensive experiments across a multitude of multimodal tasks, such as VQA (Kembhavi et al. 2016; Lu et al. 2022), hallucination evaluation (Li et al. 2023b), visual reasoning (Fu et al. 2023; Yue et al. 2023; Liu et al. 2023c), and video understanding (Fu et al. 2024; Jang et al. 2017). We use lmms-eval (Bo Li et al. 2024) and VLMEvalKit (Duan et al. 2024) to evaluate MLLMs on different datasets. In our setup, the tiny subset size was set to 20, and the threshold η was set to 0.003. For the multimodal chatbot, we withdraw vision tokens at the intermediate layers of MLLMs.

Figure 6: Dialogue comparison between baseline and VTW.

Multimodal Chatbot

We aim to boost the chatbot's response speed without compromising response quality. We compare the responses of MLLMs before and after applying VTW in Figure 6. We can see that the chatbots equipped with our VTW maintain their ability to produce the correct and similar outcomes compared to standard chatbots, despite lacking vision tokens in the deep layers. This highlights VTW's success in accelerating the chatbot's responses while holding its performance.

Quantitative Evaluation

Visual Question Answering (VQA). In VQA, MLLMs interpret images before answering questions. We test our

Methods	TFLOPs ↓	AI2D ↑	SQA_Img ↑	MMMU_Val ↑	MMB_EN ↑	POPE ↑	MME ↑
LLaVA-1.5-7B	8.48 (100.00%)	55.21	69.61	35.60	64.09	85.83	1866.10
+ FastV	4.91 (57.90%)	55.14	68.96	35.80	**64.26**	82.49	1864.35
+ Rand (K=16)	4.68 (55.19%)	4.40	9.07	29.30	0.52	82.22	139.42
+ VTW† (K = Random[8,24])	≈4.68 (≈55.19%)	55.36	69.21	36.10	55.61	76.05	1741.34
+ VTW (K=16)	**4.68** (**55.19**%)	**55.44**	**69.66**	**36.30**	64.00	**85.96**	**1872.43**
LLaVA-1.5-13B	16.50 (100.00%)	59.26	72.83	**34.90**	68.73	**86.02**	1827.26
+ FastV	9.56 (57.94%)	58.87	**73.03**	34.60	68.30	85.15	**1855.11**
+ Rand (K=20)	9.10 (55.15%)	2.49	7.93	25.80	0.94	83.78	41.74
+ VTW† (K = Random[12,28])	≈9.10 (≈55.15%)	58.35	71.99	34.20	62.02	75.06	1661.36
+ VTW (K=20)	**9.10** (**55.15**%)	**59.39**	72.88	**34.90**	**68.81**	85.93	1828.79
LLaVA-NeXT-7B	28.73 (100.00%)	65.31	**70.15**	35.30	**67.18**	**86.44**	1846.33
+ FastV	15.67 (54.54%)	64.86	68.96	**35.70**	66.84	85.98	1786.17
+ Rand (K=16)	14.80 (51.51%)	0.84	3.37	24.30	00.17	80.00	18.97
+ VTW† (K = Random[8,24])	≈14.80 (≈51.51%)	64.54	69.86	35.40	59.28	75.73	1723.65
+ VTW (K=16)	**14.80** (**51.51**%)	**65.35**	70.00	**35.70**	**67.18**	86.33	**1857.35**

Table 2: Comparison of various training-free methods for accelerating MLLMs inference. SQA_Img, MMMU_Val, and MMB_EN originate from ScienceQA (Lu et al. 2022), MMMU (Lu et al. 2022), and MMBench (Liu et al. 2023c), respectively. For a fair comparison, we manually set K as 16 to keep VTW's FLOPs lower than FastV (Chen et al. 2024). We use the average input length on MME to calculate TFLOPs. VTW† drops vision tokens in a random deep layer K. Rand randomly discards the same number of input tokens as the visual tokens. We employ **bold** formatting to highlight the best result.

VTW method on two popular VQA datasets: AI2D (Kembhavi et al. 2016) and SQA_image (Lu et al. 2022). Table 2 shows that VTW outperforms FastV and uses fewer FLOPs across different MLLMs. It also achieves lossless acceleration compared to the baseline, with nearly half the FLOPs. Notably, VTW outperforms the baseline by utilizing visual data in shallow layers and avoids excessive attention to irrelevant information that is considered noise.

Visual Reasoning. Visual reasoning demands heightened perception, knowledge, and reasoning capabilities from the model compared to VQA. We select MMMU_Val (Yue et al. 2023) and MMB_EN (Liu et al. 2023c) as our benchmarks for evaluation. The results presented in Table 2 demonstrate that VTW achieves comparable or even superior performance to the baseline and FastV, with fewer FLOPs.

Hallucination Evaluation. Hallucinations can degrade MLLMs performance and severely impact user experiences in real-world applications. We conduct experiments on POPE (Li et al. 2023b) to investigate the impact of VTW on hallucinations. As illustrated in Table 2, VTW achieves comparable performance to the baseline. Conversely, FastV, which removes vision tokens after the second layer, exacerbates MLLMs' hallucinations, suggesting that premature removal of vision tokens exacerbates MLLMs hallucinations.

Comprehensive Evaluation. MME (Fu et al. 2023) accesses both perception and cognition across a total of 14 subtasks such as OCR, object localization, and attribute recognition. In Table 2, VTW demonstrates comparable performance to the baseline in a comprehensive benchmark, showing its excellent generalization ability in fine-grained tasks.

Comparisons with Other VTW variants. In Table 2, Rand shows a significant drop in performance across all tasks, indicating that removing text tokens greatly impacts performance. VTW† shows that removing visual information too early affects performance. VTW removes vision tokens in a proper layer, leading to the best performance.

Model	FLOPs	TGIF Acc	VideoMME Overall	Avg
Video-LLaVA	100.00%	0.25	**0.30**	0.28
+ FastV	53.13%	0.21	**0.30**	0.26
+ VTW (K=16)	**50.00%**	**0.27**	**0.30**	**0.29**

Table 3: Results on Video Question Answering Tasks. We only calculate vision tokens' FLOPs.

Video Understanding. We provide the results of VTW on different video question answering tasks (TGIF (Jang et al. 2017) and VideoMMe (Fu et al. 2024)) in Table 3. VTW can generalize well in these video tasks and remain comparable to the baseline and FastV even with fewer FLOPs.

Cost Analysis

The GPU memory overhead and latency comparisons between original MLLMs, FastV and VTW($K = 16$) are presented in Table 4. We observe that during the inference of MLLMs, VTW reduces the GPU memory overhead by 35% for each sample, while FastV needs to store all image tokens' KV Cache during the first forward, leading to a high peak GPU memory. Furthermore, VTW reduces FLOPs by nearly half and the latency per sample to approximately $0.63\times$ in comparison with the baseline models.

Downstream Task

To evaluate whether VTW is still workable in pixel-level fine-grained task, we apply it to LISA (Lai et al. 2023)

Metric	Baseline	FastV	VTW
KV Cache	✓	✓	✓
Flash Attention	✓	✗	✓
(a) Model GM	14.1 G	14.1 G	**14.1 G** (**1.00**×)
(b) Peak Inference GM	17.1 G	17.1 G	**16.1 G** (**0.94**×)
(b)-(a) Per Sample GM	3.1 G	3.1 G	**2.0 G** (**0.65**×)
TFLOPs	22.9	12.9	**12.3** (**0.54**×)
Latency/Example	0.52 s	0.40 s	**0.33 s** (**0.63**×)

Table 4: The comparisons of practical GPU memory overhead and latency of LLaVa-NeXT-7B in SQA_Image on one NVIDIA RTX 3090. GM stands for GPU memory.

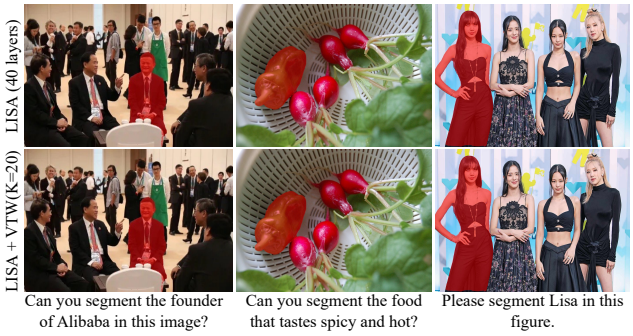

Figure 7: VTW's results on the segmentation task.

which uses MLLMs for segmentation task. As visualized in Figure 7, VTW does not degrade the segmentation ability of LISA, showing its excellent generalization ability in the fine-grained downstream task.

Ablation Studies

We conduct ablations using LLaVA-1.5-7B on SQA_Image.

Ablations on Visual Tokens Withdrawal Layers. As depicted in Figure 8(a), a small value of K, indicating an early withdrawal of vision tokens, leads to a degradation in the performance of MLLMs. When K exceeds a specific layer number, VTW performs similarly to the baseline. This observation further proves that vision tokens are unnecessary in the deep layers of MLLMs.

Ablations on Threshold η. Recall that we use KL divergence to determine the withdrawal layer K on a tiny subset of target datasets, as formulated in Eq. (6). As depicted

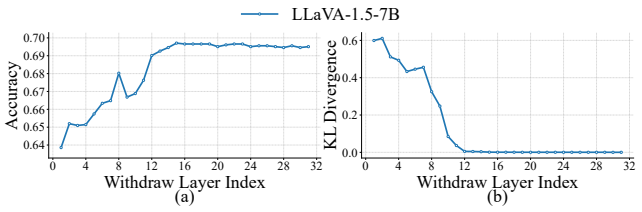

Figure 8: Ablation study on visual tokens withdrawal layer K. (a) Accuracy *v.s.* K; (b) KL divergence *v.s.* K.

η	0.006	0.005	0.004	0.003	0.002	0.001
K	12	13	14	15	15	15
Accuracy	69.01	69.26	69.46	69.70	69.70	69.70

Table 5: Ablation study on threshold η.

Position embedding	SQA_Image	MMMU_Val	POPE
LLaVA-1.5-7B	69.61	35.60	85.83
Rearrange	69.56	35.90	85.41
Keep	**69.66**	**36.30**	**85.96**

Table 6: Ablation study on position embedding. "Rearrange" and "Keep" mean to rearrange and keep the position embeddings of remaining tokens after withdrawing vision tokens.

in Figur 8(b), the KL divergence is large when K is small, and it converges as K exceeds a specific threshold. Combining the results from Figure 8 and Table 5, we observe that when the KL divergence converges, the accuracy of VTW also converges to that of the baseline. Thus, we select KL as our criterion for determining the withdrawal layer K.

Ablations on Position Embedding. LLMs use position embeddings (PE) to model the positional relationships between tokens. After withdrawing vision tokens, the PE of the remaining text tokens can either be kept as is or rearranged for contiguous positions. As shown in Table 6, "Keep" outperforms "Rearrange" in VTW across different datasets.

Limitations and Future Works

We are the first to find information migration happens in the initial layers of MLLMs. However, more complex tasks may need additional layers for this migration. Thus, the reduction in FLOPs is relatively marginal for complex tasks compared to simpler ones. Future works can try to apply VTW in the training stage to boost the information migration, which can save the training cost and further improve the performance of VTW. There is also a chance to investigate the information migration in other modalities, like audio.

Conclusion

We have introduced visual tokens withdrawal (VTW), a plug-and-play module for faster MLLMs inference. VTW is inspired by: (1) The attention sink phenomenon in LLMs persists in MLLMs. (2) The occurrence of information migration, indicates that visual information migrates to subsequent text tokens within the first few layers of MLLMs. Building upon these observations, we deduce that vision tokens become unnecessary in the deep layers of MLLMs, despite they take up a significant computational overhead. Thus, we withdraw vision tokens at specific layers in MLLMs. Experiments across various multimodal tasks and chatbots validate the efficacy of VTW in boosting MLLMs for rapid inference without compromising performance.

Acknowledgments

This work was supported by National Science and Technology Major Project (No. 2022ZD0118201), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. U23A20383, No. U21A20472, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No. 2021J06003, No.2022J06001).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 Technical Report. arXiv:2303.08774.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Bo Li, K. Z., Peiyuan Zhang; et al. 2024. LMMs-Eval: Accelerating the Development of Large Multimoal Models.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2023. Token Merging: Your ViT But Faster. In *ICLR*, 1–20.

Cao, J.; Ye, P.; Li, S.; Yu, C.; Tang, Y.; Lu, J.; and Chen, T. 2024. MADTP: Multimodal Alignment-Guided Dynamic Token Pruning for Accelerating Vision-Language Transformer. arXiv:2403.02991.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. arXiv:2403.06764.

Chen, M.; Lin, M.; Li, K.; Shen, Y.; Wu, Y.; Chao, F.; and Ji, R. 2023a. CF-ViT: A General Coarse-to-Fine Method for Vision Transformer. In *AAAI*, 7042–7052.

Chen, M.; Shao, W.; Xu, P.; Lin, M.; Zhang, K.; Chao, F.; Ji, R.; Qiao, Y.; and Luo, P. 2023b. DiffRate : Differentiable Compression Rate for Efficient Vision Transformers. In *ICCV*, 17164–17174.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 16344–16359.

Duan, H.; Yang, J.; Qiao, Y.; Fang, X.; Chen, L.; Liu, Y.; Dong, X.; Zang, Y.; Zhang, P.; Wang, J.; Lin, D.; and Chen, K. 2024. VLMEvalKit: An Open-Source Toolkit for Evaluating Large Multi-Modality Models. arXiv:2407.11691.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; Wu, Y.; and Ji, R. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; et al. 2024. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. arXiv:2405.21075.

Guo, Z.; Xu, R.; Yao, Y.; Cui, J.; Ni, Z.; Ge, C.; Chua, T.-S.; Liu, Z.; and Huang, G. 2024. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *ECCV*, 390–406.

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

Kembhavi, A.; Salvato, M.; Kolve, E.; Seo, M.; Hajishirzi, H.; and Farhadi, A. 2016. A Diagram is Worth A Dozen Images. In *ECCV*, 235–251.

Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; and Jia, J. 2023. LISA: Reasoning Segmentation via Large Language Model. arXiv:2308.00692.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. In *ICML*, 19730–19742.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, W. X.; and Wen, J.-R. 2023b. Evaluating Object Hallucination in Large Vision-Language Models. arXiv:2305.10355.

Li, Y.; Wang, C.; and Jia, J. 2023. LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models. arXiv:2311.17043.

Liang, Y.; Chongjian, G.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. EViT: Expediting Vision Transformers via Token Reorganizations. In *ICLR*, 1–21.

Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023a. Video-Llava: Learning United Visual Representation by Alignment Before Projection. arXiv:2311.10122.

Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023b. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. arXiv:2311.10122.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2023a. Improved Baselines with Visual Instruction Tuning. arXiv:2310.03744.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023b. Visual Instruction Tuning. In *NeurIPS*, 1–19.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2023c. MMBench: Is Your Multi-modal Model an All-around Player? arXiv:2307.06281.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *NeurIPS*, 2507–2521.

Marin, D.; Chang, J.-H. R.; Ranjan, A.; Prabhu, A.; Rastegari, M.; and Tuzel, O. 2021. Token Pooling in Vision Transformers. arXiv:2110.03860.

Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Heek, J.; Xiao, K.; Agrawal, S.; and Dean, J. 2023. Efficiently Scaling Transformer Inference. In *MLsys*, 1–18.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 8748–8763.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient Vision Transformers with Dynamic Token Sparsification. In *NeurIPS*, 13937–13949.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. arXiv:2403.15388.

Shi, D.; Tao, C.; Rao, A.; Yang, Z.; Yuan, C.; and Wang, J. 2023. Crossget: Cross-guided Ensemble of Tokens for Accelerating Vision-Language Transformers. arXiv:2305.17455.

Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: A Family Of Highly Capable Multimodal Models. arXiv:2312.11805.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wu, X.; Zeng, F.; Wang, X.; Wang, Y.; and Chen, X. 2023. PPT: Token Pruning and Pooling for Efficient Vision Transformers. arXiv:2310.01812.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2023. Efficient Streaming Language Models with Attention Sinks. arXiv:2309.17453.

Xu, Y.; Zhang, Z.; Zhang, M.; Sheng, K.; Li, K.; Dong, W.; Zhang, L.; Xu, C.; and Sun, X. 2022. Evo-Vit: Slow-Fast Token Evolution for Dynamic Vision Transformer. In *AAAI*, 2964–2972.

Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; et al. 2023. Mmmu: A Massive Multi-Discipline Multimodal Understanding and Reasoning Benchmark for Expert Agi. arXiv:2311.16502.