Multimodal Alignment and Acceleration of Large Models

June 6, 2025

1 Introduction

In recent years, the rapid evolution of Large Multimodal Models (LMMs) has significantly propelled breakthroughs in artificial intelligence across tasks such as vision-language understanding, image-text generation, visual question answering, and multimodal dialogue. This progress has been facilitated by the accessibility of massive cross-modal datasets, the widespread adoption of Transformer architectures, and the continuous evolution of the pre-training-fine-tuning paradigm.

Among these, Vision-Language Models (VLMs), as a core branch within multimodal research, have become a key focus in both academia and industry. The goal of VLMs is to model the semantic relationships between images and natural language, enabling machines to achieve effective image-text understanding, description generation, and cross-modal reasoning in complex tasks. Over the past few years, VLM architectures have evolved from early image captioning models to pre-trained representative frameworks like CLIP[15] and ViLBERT[14], and further to more recent, versatile large-scale integrated models such as BLIP[9], Flamingo[1], and LLaVA[12]. This evolution marks a gradual transition from task-specific designs towards a general large-model paradigm. Notably, the contrastive learning paradigm represented by CLIP[15], and the image-text instruction alignment paradigm exemplified by Flamingo[1] and LLaVA[12], have pioneered new pathways for building vision-language models with strong generalization capabilities. Through joint pre-training on large-scale image-text pair datasets, these models demonstrate exceptional multimodal understanding and reasoning abilities, enabling zero-shot or few-shot transfer across various downstream tasks.

The rapid development of VLMs has intensified research focus on a pivotal challenge: inference acceleration. Although employing larger visual encoders, more complex visual projection layers, or larger LLMs can significantly enhance model performance, their inference efficiency, parameter scale, and deployment costs have become critical bottlenecks in practical applications. Research on inference acceleration holds significant potential for facilitating real-world applications of VLMs, particularly in enabling their deployment on mobile platforms.

To obtain efficient VLMs, research efforts primarily follow two paths: One approach investigates leveraging lighter visual encoders and smaller LLMs to reach the performance of large-scale VLMs, as seen in works like Bunny[6], MobileVLM[4], and FastVLM[19]. These small-parameter models enable deployment on users' mobile devices. The other approach concentrates on better understanding the fusion of visual and textual information to accelerate model inference. These works such as FastV[3] and FitPrune[21] primarily focus on analyzing attention allocation between visual and text tokens within LLMs, subsequently accelerating visual information flow through transformer layer pruning. However, this strategy is fundamentally flawed for two reasons: (1) the brute-force pruning approach inevitably causes irreversible information loss, and (2) the attention analytical findings presented in these studies cannot be universally generalized across all VLMs.

From a cross-modal alignment perspective, it is intuitive that if the vision encoder and projector effectively abstract visual information, the large language model (LLM) requires minimal effort to integrate different modalities. This efficiency largely depends on the degree of multimodal alignment. In current VLMs, particularly light-weight ones, various adaptations of classical vision encoders are employed, alongside complex projectors designed to fuse information and reduce the number of image tokens.

Therefore, we argue for a systematic investigation into how different VLM components affect the cross-attention mechanisms in LLMs, especially small size VLMs. Variables include the size of vision encoder, design of projector and the size of LLM. This analysis should inform the design of optimized dynamic attention allocation strategies to achieve more generalized and efficient VLM acceleration. Thus, the main contributions of this work are:

- We identify and experimentally validate critical limitations in existing VLM acceleration approaches based on attention allocation.
- We conduct comprehensive analysis of the vision encoder, projector, and LLM, investigating how visual information processing of each module affects attention allocation in the VLM(especially for light-weight VLMs).
- Given valuable observations, our study provides principled guidelines for designing more effective acceleration framework.

2 Related Work

2.1 Light-weight VLMs

Recent efforts in light-weight vision-language modeling primarily leverage under 3B parameter LLM backbones, compact vision encoders, and streamlined vision-to-language projectors to dramatically reduce inference costs. For instance, MobileVLM[4] and LLaVA-Phi[25] both use CLIP ViT-L/14[15] as the visual encoder and apply LDP or MLP projection layers to connect low resolution (336 px) image features with small LLMs (MobileLLaMA or Phi-2, each has 2.7 B parameters), achieving roughly five-fold FLOPS and memory reductions compared to 13 B-scale counterparts. A similar backbone-centric strategy is seen in Bunny[6], which couples a SigLIP-SO[23] encoder with Phi-2 via an MLP projector to retain competitive accuracy on VQA and related benchmarks while maintaining a light compute footprint. Going a step further, TinyGPT-V[22] integrates an EVA[17] vision encoder and a Q-Former[8] module to prune visual tokens before feeding them into Phi-2, thereby lowering overall FLOPS without appreciable degradation in downstream task performance. DeepSeek-VL[13] exemplifies the extreme of this paradigm by pairing a SigLIP-L[23] encoder with a 1.3 B DeepSeek-LLM[5] and a basic MLP projector, making it one of the lightest open-source VLMs focused on rapid multimodal retrieval and VQA.

2.2 Multimodal alignment in VLMs

In cross-modal alignment for VLMs, a common approach involves leveraging pre-trained visual encoders, such as CLIP's Vision Transformer(ViT)[15], to extract rich visual features from images. These features are then projected into the textual embedding space of a LLM using a trainable projector module, often implemented as a MLP. This two-step extraction using pre-trained vision encoders followed by modality specific projection has become a standard framework for bridging vision and language in VLMs.

2.3 Inference acceleration based on attention allocation

In the realm of inference acceleration for Vision-Language Models (VLMs) based on attention allocation mechanisms, several innovative approaches have emerged. FasterVLM[24] identifies that traditional text-visual attention in language models suffers from positional bias and dispersion, leading to suboptimal token pruning. Instead, it leverages visual encoder attention to select important tokens and removes redundant ones via similarity. FastV[3] focuses on the inefficient attention phenomenon in deep layers, pruning visual tokens with low attention scores after early layers (e.g., layer 2) based on average attention allocation. FitPrune[21] frames token pruning as a distribution-fitting problem, minimizing the divergence between attention distributions before and after pruning. By jointly considering self- and cross-attention, it generates optimal pruning strategies in minutes using small-scale data.

In fact, these acceleration methods have only been experimentally validated on 7B-scale models, primarily limited to the LLaVA[12] series (including both v1.5[10] and v1.6[11] variants). While these LLaVA models share highly consistent training data, architectural designs, and component modules, our analysis reveals that the observed findings do not consistently generalize to other fundamentally different VLMs, particularly light-weight efficient VLMs. And brute-force approaches that uniformly prune visual tokens after the earlier layers would no longer be applicable. If such methods are still employed, the resulting loss of visual information may fail to achieve an optimal trade-off with inference acceleration. Given that effective visual-textual information fusion requires the coordinated participation of vision encoders, projectors, and LLMs, we argue that more comprehensive experimentation is necessary to develop a truly generally effective inference acceleration framework.

3 Preliminaries

3.1 Architecture of VLMs

As shown in Fig1, the standard VLM architecture comprises three core components: a vision encoder, a projector module, and a large language model. Among prominent VLMs, LLaVA ([12]) exemplifies this established paradigm. Its architecture employs CLIP-ViT [15] as the vision encoder for extracting grid-wise image features, which are subsequently transformed by a lightweight projector (typically a linear layer or shallow MLP) into visual tokens compatible with the LLM's embedding space. These aligned tokens are then concatenated with text embeddings and processed by a LLM backbone for multimodal reasoning.

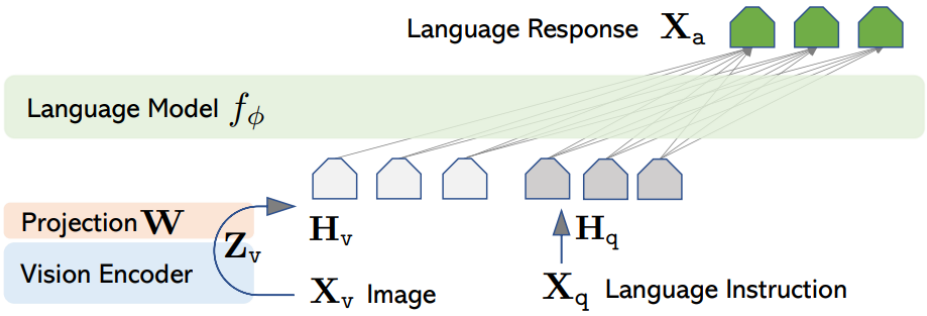

Figure 1: Standard architecture of VLMs

3.2 Vision encoder

VLMs predominantly utilize pre-trained vision encoders aligned with textual semantics to facilitate cross-modal feature integration. CLIP-ViT[15] has emerged as the standard in this context, leveraging its established cross-modal alignment capabilities to bridge visual and linguistic domains. A competitive alternative, SigLIP[23], enhances efficiency and performance through an optimized loss function, often serving as a complement or replacement for CLIP in models such as Bunny[6] and Idefics2[7]. Besides, EVA-CLIP[17] exhibits superior parameter efficiency with excellent performance. The field increasingly favors hybrid light-weight encoder designs, such as FastVLM[19], which employs a hybrid architecture of convolutional blocks and Transformer layers. These designs strike a balance between perceptual granularity and computational overhead, making them well-suited for resource-constrained scenarios where efficiency is critical.

3.3 Projectors

Projector modules play a crucial role in VLMs by bridging visual encoders with LLMs. For example, the lightweight downsample projector (LDP) used in MobileVLM[4] combines pointwise convolutions to adjust channel dimensions, spatial pooling to shrink token counts, and a residual positional encoding generator to inject spatial structure—yielding up to 99.8 % fewer parameters and dramatically lower FLOPS than a standard fully connected mapping. A Perceiver-style Resampler[1] adopts a small set of learnable latent queries that perform cross-attention over the entire sequence of visual tokens, distilling hundreds of inputs into a fixed number of summary vectors so the LLM receives a consistent-length representation. In contrast, simple MLP projectors—seen in models like LLaVA[12]—employ one or more fully connected layers (often with nonlinear activations) to map frozen visual embeddings (e.g., from CLIP[15] or SigLIP[23]) directly into the LLM's token space, relying on the assumption that the input features are already semantically rich.

4 Methodology

To rigorously analyze the attention mechanisms, we first formally define attention allocation in the inference phase. Let α represents the distribution of each output tokens' attention score during the decoding process of one response. For the i-th token in the j-th layer, we define $\alpha_{sys}^{i,j}$, $\alpha_{img}^{i,j}$, $\alpha_{inst}^{i,j}$, and $\alpha_{out}^{i,j}$ to represent the total attention scores allocated by the current token to the system prompt, image tokens, user instructions (e.g., queries about the image), and output tokens, respectively. Obviously, we have:

$$\alpha_{sys}^{i,j} + \alpha_{img}^{i,j} + \alpha_{inst}^{i,j} + \alpha_{out}^{i,j} = 1 \tag{1}$$

Then we can compute the total attention allocation λ to mark the total attention score that one type of tokens(sys, img, inst or out) received in one layer.

$$\lambda_{\mathrm{T}}^{j} = \sum_{i=1}^{n} \alpha_{\mathrm{T}}^{i,j}, \quad \mathrm{T} \in \{sys,\, img,\, inst,\, out\} \tag{2}$$

where n is the total number of tokens in generated response.

Finally, the attention allocation α is averaged across all attention heads in the LLMs. Following related works[3][24], we adopt three ways to investigate the allocation mechanism of attention across different modalities in LLM backbones, thereby validating our prior hypotheses.

4.1 Attention heatmap

We present VLMs with an image and a corresponding query, capturing the attention scores within the LLM backbone during response generation. By segmenting the input sequence into system tokens, image tokens, instruction tokens, and output tokens, we can directly analyze the attention distribution across these components by visualizing them through a heatmap.

4.2 Attention efficiency

To analyze the attention allocation mechanism at a finer token-wise granularity, based on above attention hearmap, we quantify the proportion of attention allocated to the four distinct token types. The attention efficiency ϵ is defined to represent the average attention score per type's token received in one layer during process of one response.

$$\epsilon_{\mathrm{T}}^{j} = \frac{\sum_{i=1}^{n} \alpha_{\mathrm{T}}^{i,j}}{|\mathrm{T}|}, \quad \mathrm{T} \in \{sys,\, img,\, inst,\, out\} \tag{3}$$

where $|\mathrm{T}|$ is the number of all tokens of type T, n is the total number of tokens in generated response.

This metric is primarily employed in subsequent experiments to quantify the overall attention received by each token type.

4.3 Attention Shift

Building on the aforementioned methods, we focus on the attention allocation to image tokens. Specifically, we investigate the attention distribution within these tokens to examine the overall attention scores assigned by the response to each image token. Let we define $d\alpha_i^j$ as the sum of attention scores that each token in response paid on the i-th token in image tokens within j-th layer. We have:

$$\lambda_{img}^{j} = \sum_{i=1}^{m} d\alpha_i^j \tag{4}$$

where m is the total number of image tokens. Here the final $d\alpha$ is also averaged cross all attention heads in LLMs.

While this analysis is not central to understanding the attention allocation mechanism, it provides valuable insights into how attention is distributed across different positions within the image token sequence.

5 Experiment

In this section, we conduct a comprehensive experimental analysis on various visual-language models (VLMs). For each model, an image-query pair is provided to generate a response. During inference, we record the attention scores and apply the aforementioned methods to analyze them.

5.1 Settings

We use the default image and query introduced in LLaVA[12] shown in Appendix8.1.

To conduct a comprehensive experiment, we survey current open-source VLMs with parameter sizes under 8 billion, with a particular focus on light-weight models. Based on the aforementioned methodologies, we compile a collection of VLMs in Table1, encompassing various components of VLMs with respect to their size and type.

The entire experiment is conducted on an NVIDIA A100 40G GPU, utilizing the Transformers library. The temperature is 0.2 and max response length is limited to 512.

Table 1: Overview of Open-Source Visual-Language Models (VLMs) Under 8B Parameters in Our Experiment. Parameter sizes are rounded to one decimal place.

Model	**Vision Encoder**		**Projector**	**LLM**	
	Variants	**Parameter Size (B)**		**Variants**	**Parameter Size (B)**
TinyGPT-V[22]	EVA + MLP	0.7	Q-Former	Phi-2	2.7
Qwen2-VL-2B[20]	ViT	0.7	PatchMerger	Qwen2	1.5
Qwen2-VL-7B[20]	ViT	0.7	PatchMerger	Qwen2	7.6
Idefics2-8B[7]	SigLIP-SO400M	0.4	MLP+Resampler	Mistral	7.3
Qwen2-5-VL-3B[2]	ViT	0.7	PatchMerger	Qwen2.5	3
Qwen2-5-VL-7B[2]	ViT	0.7	PatchMerger	Qwen2.5	7
Mobilevlm-v2-7B[4]	CLIP-L/14	0.3	LDP	Vicuna	7
Bunny-phi-2-siglip[6]	SigLIP-SO400M	0.4	MLP	Phi-2	2.7
Bunny-Phi-2-eva[6]	EVA02-CLIP-L	0.4	MLP	Phi-2	2.7
Bunny-phi-1.5-siglip[6]	SigLIP-SO400M	0.4	MLP	Phi-1.5	1.3
Bbunny-stablelm-2-eva[6]	EVA02-CLIP-L	0.4	MLP	Stablelm-2-1.6B	1.6
Bunny-LLaVA-Phi2[6]	CLIP-L/14	0.3	MLP	Phi-2	2.7
LLaVA-v1.5-vicuna-7B[10]	CLIP-L/14	0.3	MLP	Vicuna	7
LLaVA-v1.6-mistral-7B[11]	CLIP-L/14	0.3	MLP	Mistral	7.3
LLaVA-v1.6-vicuna-7B[11]	CLIP-L/14	0.3	MLP	Vicuna	7
FastVLM-0.5B[19]	FastViTHD	0.1	MLP	Qwen2	0.5

5.2 Reproduction

In this section, we first replicate the findings of FastV[3] in Figure2, which employs LLaVA-v1.5 to analyze attention heatmaps.

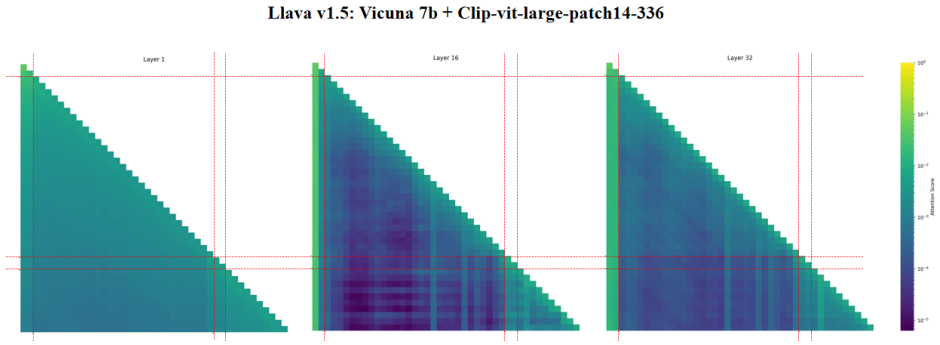

Figure 2: Reproduction of attention map on LLaVA v1.5

We can draw conclusions from the figure that are consistent with FastV[3]: in deeper layers, the attention allocation on image tokens becomes sparse, thus validating the findings and providing the potential for accelerating inference by pruning image tokens.

5.3 Experiment design

Based on the previous introduction, we posit that the allocation of cross-modal attention in VLMs is influenced by the vision encoder and projector's ability to abstract visual information, with the LLM's comprehension capability also playing a critical role. Intuitively, we approximate a model's inherent capability to its parameter size: generally, models with larger parameter sizes tend to achieve high performance, as a larger vision encoder can extract fine-grained representations, and a larger LLM can exhibit stronger comprehension abilities. Building on these assumptions, we design a series of experiments, which are categorized into three aspects: the impact of vision encoder size on attention allocation, the influence of LLM size on attention allocation, and the effect of projector type on attention allocation.

To provide a clear illustration of the attention allocation mechanism, our experimental results focus on reporting the obtained attention efficiency, instead of displaying extensive heatmaps for a qualitative assessment of the sparsity in image tokens' attention.

Here, we provide a brief introduction to the complex vision encoder and projector mentioned in Table1.

Vision Encoders

- CLIP-L/14[15]: a variant of the CLIP, featuring a vision transformer (ViT) architecture with a patch size of 14. It consists of a large-scale vision encoder designed to process images by dividing them into 14x14 pixel patches, paired with a text encoder to enable cross-modal understanding.
- SigLIP-SO400M[23]: an optimized variant of the SigLIP model, designed for efficient cross-modal learning with a vision encoder based on a scaled ViT architecture. It incorporates a specific setup with a 400M parameter scale, optimized for processing images and aligning them with text through contrastive learning, enhancing performance in tasks such as image-text matching.
- EVA02[17]: a scalable vision foundation model based on a plain Vision Transformer (ViT) architecture, optimized through advanced pre-training strategies such as masked image modeling (MIM) and contrastive learning, enabling strong performance across diverse downstream tasks
- FastViTHD[19]: a hybrid architecture consisting of five stages: the first three stages utilize RepMixer blocks, while the last two adopt multi-head self-attention blocks, with downsampling techniques applied to reduce the number of visual tokens. Its key advantages include lower latency in high-resolution image encoding, fewer generated tokens, and enhanced efficiency and accuracy for vision-language models, demonstrating superior precision-latency trade-offs across multiple benchmarks.

Projectors

- Q-Former[8]: a query-based vision-language projection module that employs a fixed set of learnable query tokens to extract semantic representations from visual features.
- PatchMerger[20, 2]: a lightweight module designed to reduce the number of visual tokens by merging image patches between intermediate layers, thereby decreasing computational overhead. In Qwen2-VL and Qwen2.5-VL, it operates in conjunction with a dynamic resolution mechanism, enabling the model to process images of varying resolutions efficiently.
- LDP[4]: designed for resource-constrained environments, aiming to reduce the number of visual tokens through efficient downsampling. This approach minimizes computational resource consumption while maintaining the integrity of visual semantics.
- Resampler[7]: a vision-language projector in Idefics[7] that compresses high-dimensional image features into fixed-length representations, facilitating alignment with language models. It employs efficient downsampling strategies to preserve essential visual information while enabling effective multimodal fusion.

5.4 Main results

We report the experimental results in this section to support and summarize several key conclusions. First, we give an overall look of image tokens' attention efficiency ϵ_{img} in Figure3 on all models mentioned above.

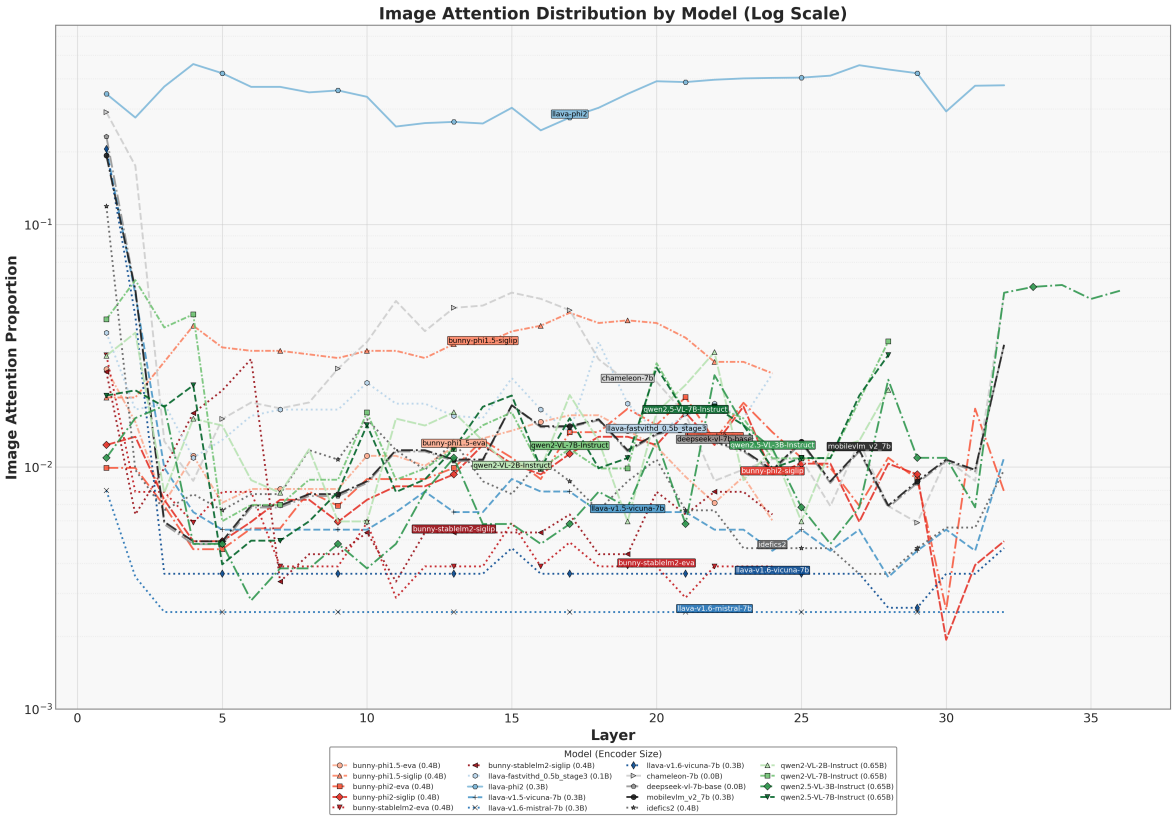

Figure 3: Image attention efficiency of all models

From the visualization, we observe that LLaVA v1.6 rapidly absorbs the information from visual tokens within the first five layers(i.e., it allocates almost no further attention to the visual modality in the subsequent layers). Although LLaVA v1.5 shows some fluctuation in attention distribution in later layers, its overall attention to the visual modality also drops to a low level after the initial few layers as analyzed in the FastV[3]. This validates the effectiveness of FastV's approach in tolerating moderate performance degradation. However, we note that many other light-weight VLMs do not exhibit such attention patterns.

In particular, models such as FastVLM-0.5B, bunny-phi1.5-siglip, and LLaVA-phi2 demonstrate a distinctive behavior: visual information flows consistently and extensively across all layers of the LLM. Even other models, in comparison to LLaVA v1.5 and v1.6, tend to maintain a higher overall level of attention efficiency with respect to visual tokens. These results suggest that directly pruning image tokens as a means to accelerate inference is not a generally effective strategy, especially for lightweight and efficient VLMs.

We argue that the reason LLaVA v1.5 and v1.6 achieve such attention sparsity lies in two major factors: first, their use of a powerful visual encoder (CLIP-L/14) that abstracts and compresses visual information effectively; second, the availability of well-curated modality alignment data, which allows the model to learn effective visual-text alignment via a dedicated alignment MLP. This enables the LLM to rely less on its own capacity to interpret visual semantics. In contrast, other models suffer from weaker understanding due to their lightweight architecture, and their design typically prioritizes reduced inference cost over modality alignment capability(for example, through the use of complex projectors that aggressively reduce the number of image tokens).

Subsequently, we perform experiments on models with identical LLM sizes to control for language model capacity, thereby ensuring comparability in their understanding ability. The results on 3B LLMs are shown in Figure4. More results on other sizes are displayed in Appendix8.2.

We observe that when the LLM parameter size is fixed at 3B, there exists substantial variance in attention efficiency across different token types among models. This indicates that it is difficult to identify a unified

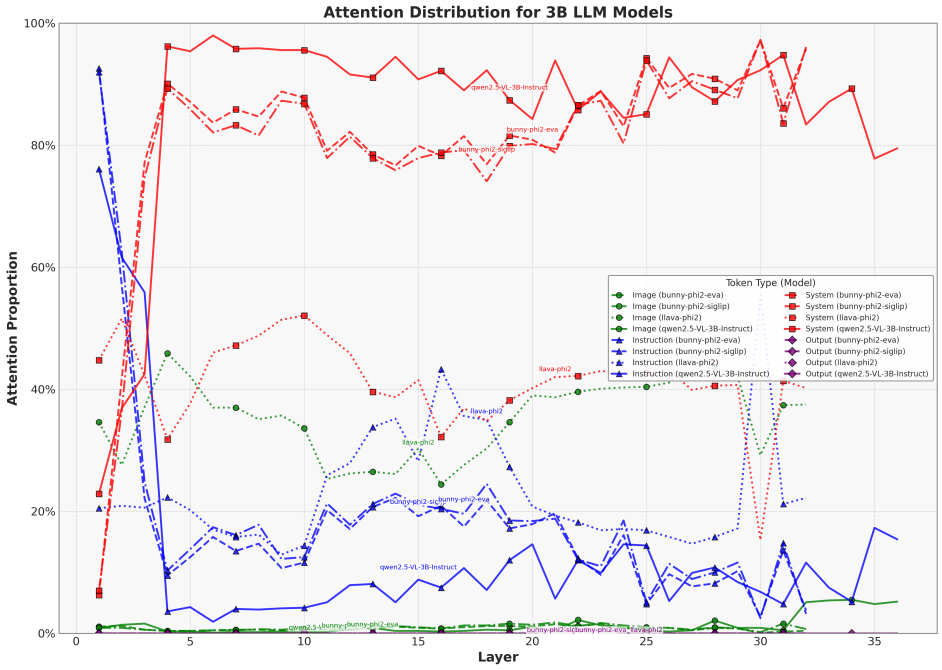

Figure 4: Attention efficiency of different token type on all 3B-LLM models

pattern for analyzing the attention allocation mechanism. Consequently, even when the LLMs have comparable capacities, other components—such as the vision encoder and the projector—can significantly affect how attention is distributed.

This influence is exemplified by models like LLaVA-phi2, where suboptimal modality alignment in the early stages leads to the LLM having to consistently allocate a greater proportion of attention to integrating visual and textual information in later layers.

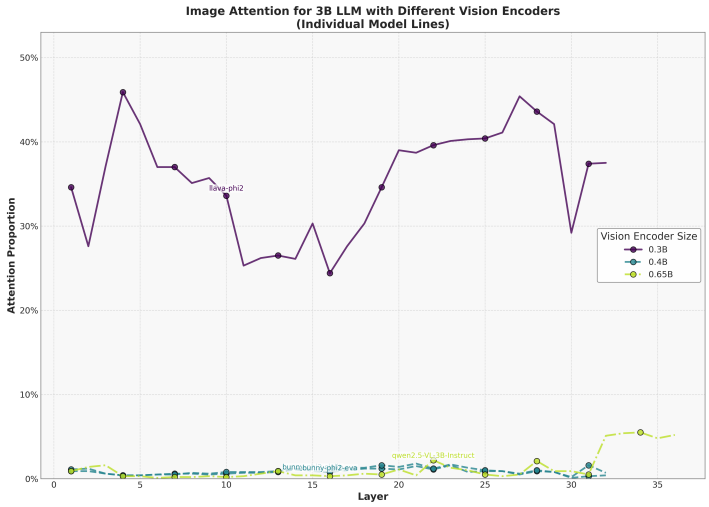

Figure 5: Image attention efficiency on different vision encoders for 3B-LLM models

Furthermore, we analyze how the capacity of the pre-trained vision encoder, as reflected by its parameter size, affects the attention allocation mechanism in Figure5. It is observed that larger vision encoders, which generally imply more powerful visual representation capabilities, facilitate more effective alignment between visual embeddings and the textual semantic space. Consequently, the LLM requires only shallow processing to incorporate visual information.

Further experiments are conducted to demonstrate how the size of the LLM influences the attention alloca-

tion to image tokens, which can be interpreted as the LLM's capability to manage image attention. As shown

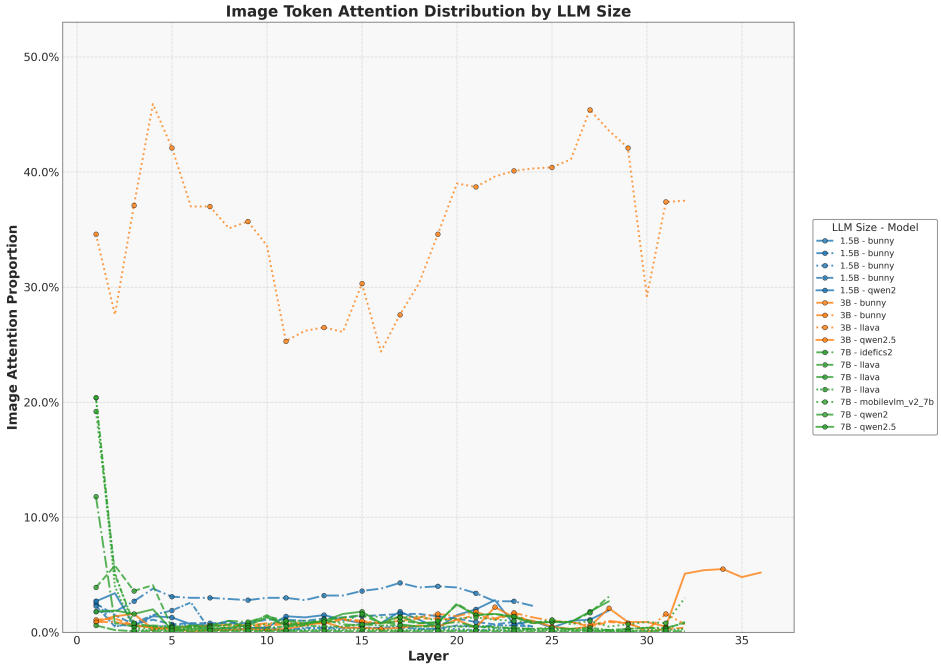

Figure 6: Image attention efficiency on different LLM size

in Figure6, excluding the outlier LLaVA-Phi2, we observe that VLMs built upon LLMs of different sizes(1.5B, 3B, and 7B) exhibit generally consistent attention allocation patterns. Specifically, no significant differences in attention distribution can be attributed to model size. Across all models, the attention allocated to image tokens is relatively high at the beginning, then gradually decreases and stabilizes at a baseline level. Therefore, we argue that the size of the LLM, or more broadly, its general language understanding capability, is not a major factor influencing attention allocation to image tokens.

As a supplementary experiment in Figure7, we applied an attention shift analysis on LLaVA v1.6 to further support our hypothesis. The visualization reveals that image tokens closer to the output tokens in the sequence tend to receive more attention, likely due to the positional bias inherent in the attention mechanism that tokens positioned closer are more likely to be attended to. This finding highlights that naively pruning image tokens at shallow layers is suboptimal. Instead, it actually suggests a more informed approach: leveraging token-level attention patterns to identify and prune only those tokens that are truly unnecessary.

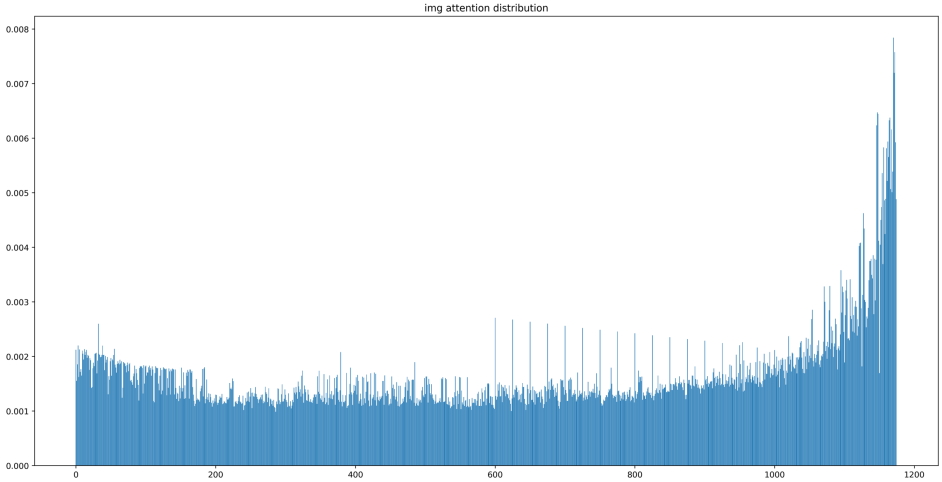

Figure 7: Attention shift phenomenon in image tokens

Finally, we examine the effect of projector complexity on attention allocation, as shown in Figure 8, while keeping the LLM size fixed at 7B to ensure a fair comparison.

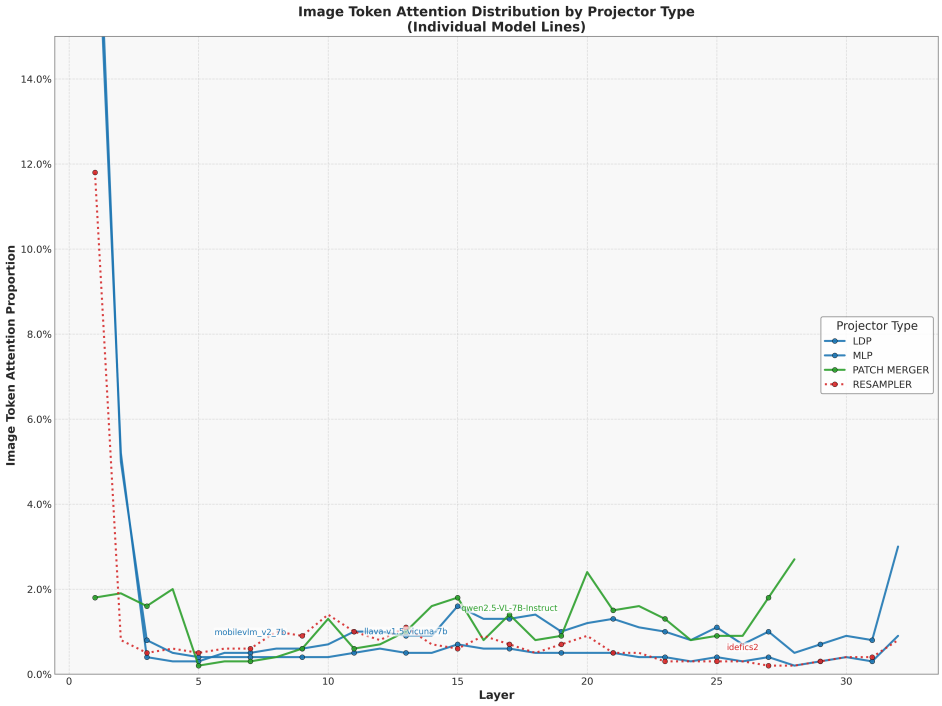

Figure 8: Image efficiency on different projectors

The results show that projector complexity and design also significantly influence attention allocation across layers. Projectors such as MLP and LDP aggressively compress visual information in early layers, resulting in sparse visual attention in deeper layers. Resampler, combined with MLP, also demonstrates a comparable effect—efficiently condensing visual information early on, which leads to reduced attention to image tokens in the deeper layers of the LLM. An interesting observation is that in deeper layers of large language models, the LDP and Resampler exhibit more sparse attention patterns compared to MLP, a phenomenon that has been overlooked in previous works. In contrast, PatchMerger retain a more consistent visual flow throughout the LLM, reflecting weaker early fusion and higher integration burden on the LLM. These findings suggest that lightweight projector designs can reduce visual token redundancy early, while more balanced designs like PatchMerger preserve fine-grained information at the cost of sustained attention demand.

Another thing: results in Figure3 also include a model which is not introduced in above sections. That is Chameleon[18], an early-fusion multimodal large language model that can jointly understand and generate text, images, and even code, with key innovations including a unified token representation for all modalities and an early-fusion architecture that enables joint training in a shared token space from the input stage. Unlike the standard three-part VLM architecture we focus on, this model adopts a different paradigm by abandoning early-stage modality fusion and instead allowing all modalities to be jointly processed within the LLM. Interestingly, results show that in Chameleon, image tokens receive increasing attention in the middle layers of the LLM, which then gradually decreases toward the final layers. This is an interesting observation that may inspire further investigation. However, as this model lies outside the scope of the VLM framework we primarily study, we do not provide a detailed analysis here.

6 Conclusion

In this work, we focus on the problem of inference acceleration for vision-language models (VLMs), raising a critical question: Are existing acceleration methods based on direct pruning truly effective and generalizable? While many prior studies aim to reduce model size to enable edge deployment, we take a different perspective. By conducting a thorough investigation into attention allocation behaviors under various influencing factors(vision encoders, projectors and LLMs) in lightweight VLMs, our study offers valuable insights

toward designing more principled and broadly applicable frameworks for VLM inference acceleration.

7 Feasible Way

Building upon our findings that the modality fusion effects introduced by different modules significantly influence the subsequent attention allocation within the LLM, we propose a feasible solution: a routing mechanism that dynamically allocates attention computation for image tokens. Specifically, during VLM training, we introduce a router module along with an auxiliary loss, enabling the model to learn which visual tokens require attention computation at each transformer layer. This allows the model to selectively skip certain tokens from processing in specific layers. Our approach is inspired by the Mixture-of-Depth[16] framework, which is a transformer-based framework that dynamically allocates compute resources by selectively routing tokens to either standard self-attention/MLP blocks or residual connections, enforcing a fixed FLOP budget while maintaining static computation graphs.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[3] 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, 2024.

[4] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and Chunhua Shen. Mobilevlm : A fast, strong and open vision language assistant for mobile devices, 2023.

[5] DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism, 2024.

[6] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective, 2024.

[7] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024.

[8] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[9] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

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

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

[12] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[13] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024.

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[16] David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models, 2024.

[17] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[18] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025.

[19] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models, 2025.

[20] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

[21] 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, 2024.

[22] Zhengqing Yuan, Zhaoxu Li, Weiran Huang, Yanfang Ye, and Lichao Sun. Tinygpt-v: Efficient multimodal large language model via small backbones, 2024.

[23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[24] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms, 2025.

[25] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multimodal assistant with small language model, 2024.

8 Appendix

8.1 Template in inference

We use default template given in LLaVA's repo and the query is appropriately extended.

> Q: What are the things I should be cautious about when I visit this place? Are there any dangerous areas or activities I should avoid? Or any other important information I should know?

8.2 More experimental results

Figure 9: Example image with resolution 336 px

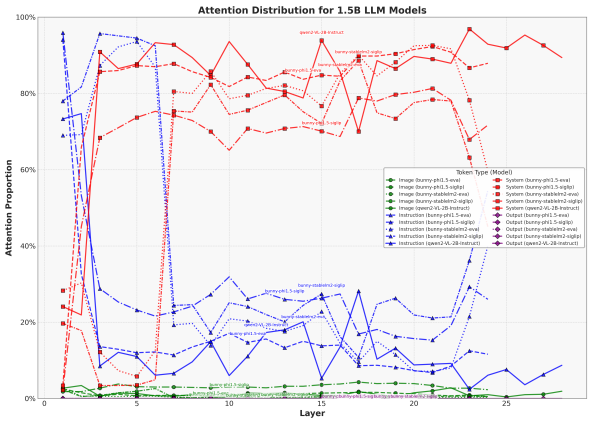

Figure 10: Attention efficiency of different token type on all 1.5B-LLM models

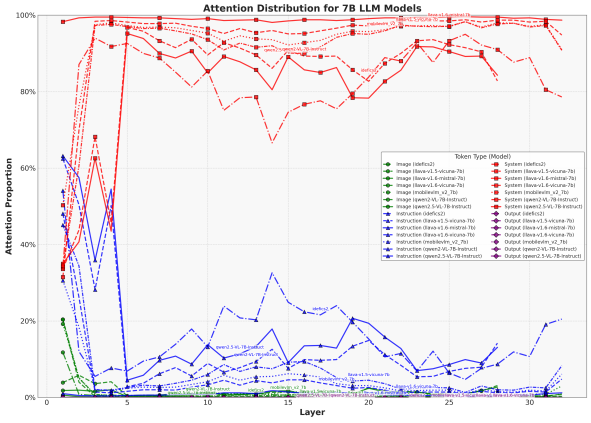

Figure 11: Attention efficiency of different token type on all 7B-LLM models

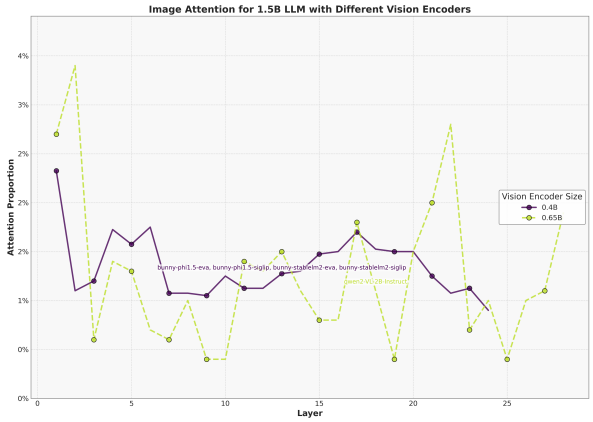

Figure 12: Image attention efficiency on different vision encoders for 1.5B-LLM models

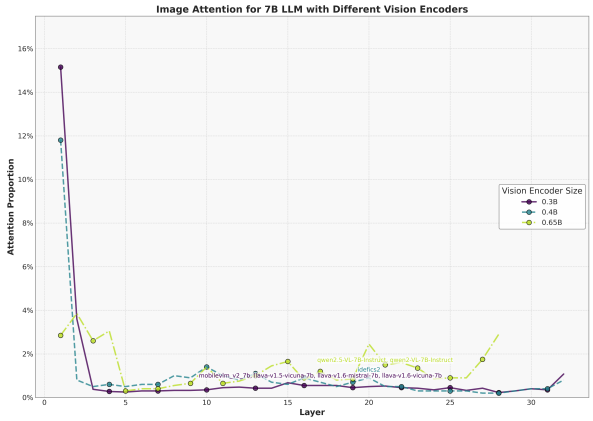

Figure 13: Image attention efficiency on different vision encoders for 1.5B-LLM models

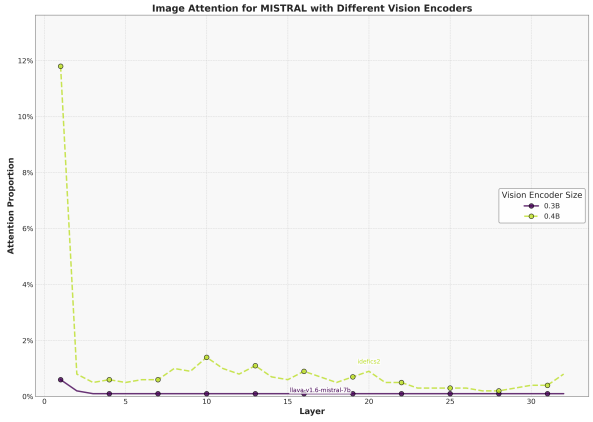

Figure 14: Image attention efficiency on different vision encoders for same LLM model(Mistral)

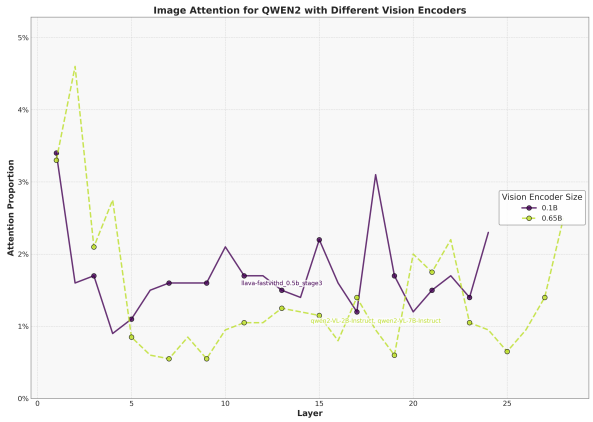

Figure 15: Image attention efficiency on different vision encoders for same LLM model(Qwen2)

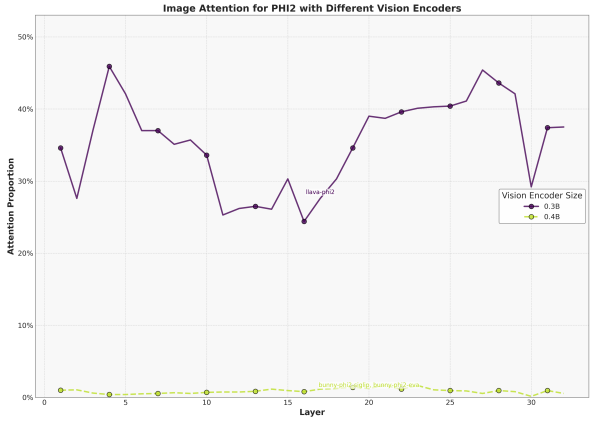

Figure 16: Image attention efficiency on different vision encoders for same LLM model(Phi2)