Parallel Vision Token Scheduling for Fast and Accurate Multimodal LMMs Inference

Wengyi Zhan[1], Mingbao Lin[2], Zhihang Lin[1,3], and 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] Rakuten, Singapore.
[3] Shanghai Innovation Institute, China.
zhanwy@stu.xmu.edu.cn, linmb001@outlook.com
lzhedu@foxmail.com, rrji@xmu.edu.cn

Abstract. Multimodal large language models (MLLMs) deliver impressive vision-language reasoning, but suffer steep inference latency because self-attention scales quadratically with sequence length and thousands of visual tokens contributed by high-resolution images. Naively pruning less-informative visual tokens reduces this burden, yet indiscriminate removal can erase contextual cues essential for background or fine-grained questions, undermining accuracy. In this paper, we present ParVTS (Parallel Vision Token Scheduling), a training-free scheduling framework that partitions visual tokens into subject and non-subject groups, processes them in parallel to transfer their semantics into question tokens, and discards the non-subject path mid-inference to reduce computation. This scheduling reduces computational complexity, requires no heuristics or additional modules, and is compatible with diverse existing MLLM architectures. Experiments across multiple MLLM backbones show that ParVTS prunes up to 88.9% of visual tokens with minimal performance drop, achieving 1.77× speedup and 70% FLOPs reduction. Our code is released at https://github.com/CrispyFeSo4/ParVTS.

Keywords: MLLMs · Efficient inference · Token scheduling

1 Introduction

Multimodal large language models (MLLMs) [24, 29, 44] with large language models (LLMs) [13, 36], which are fine-tuned for instruction following, have significantly enhanced capabilities in vision-language tasks, including complex reasoning and visual understanding. However, these benefits come at a steep computational cost.

A primary source of this cost is the quadratic complexity of the transformer's self-attention mechanism [41], which becomes prohibitive as sequence length increases. In MLLMs, visual tokens from high-resolution images often dominate the sequence—sometimes numbering in the thousands—far exceeding textual

⋆ Corresponding author.

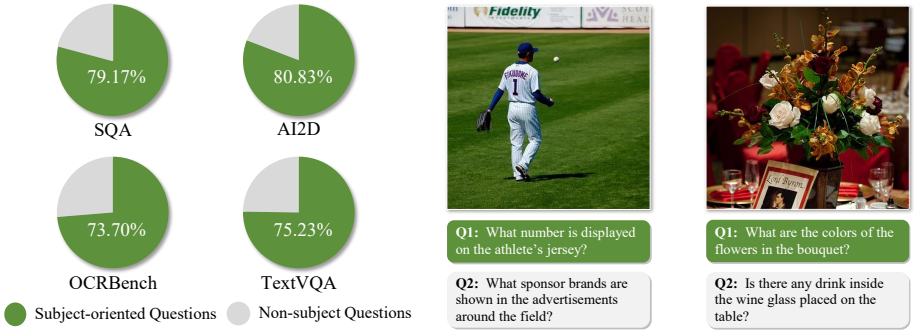

Fig. 1: Subject-oriented question distribution and examples across VQA Datasets [21, 33, 34, 38]. Left: Percentage of questions focused on subject content, with annotation details in supplementary material (Sec.7). Right: Visual examples contrasting subject-relevant (Q1) and non-subject (Q2) questions.

tokens. This imbalance greatly prolongs inference latency, posing serious challenges in latency-sensitive applications such as visual question answering (VQA) and mobile augmented reality, where real-time response is required [8].

To address this issue, prior works [3,7,9,28,37] have proposed reducing redundant visual tokens. A key observation is that only a small subset—often referred to as *subject tokens*—is necessary to answer most visual questions. Retaining these critical tokens allows inference with reduced complexity $\mathcal{O}(L_s^2)$ compared to $\mathcal{O}(L^2)$ for the full sequence, where $L_s \ll L$.

Nonetheless, while a majority of visual questions center around subject entities, a non-negligible portion of queries target background context, fine-grained details, or peripheral objects—elements often represented by *non-subject tokens*. As illustrated in Fig. 1, subject-related questions account for approximately 73% to 80% of all queries across four representative VQA datasets (SQA [34], AI2D [21], OCRBench [33], and TextVQA [38]), leaving 19% to 27% as non-subject-oriented. These non-subject questions, though fewer, often require reasoning about subtle attributes (*e.g.*, brand names on signage or the presence of transparent objects) that lie outside the core subject region.

The visual examples on the right of Fig. 1 highlight this distinction. Identifying the jersey number (Q1) requires only subject tokens, whereas recognizing the sponsor logos around the field (Q2) depends on peripheral visual information. While flower color (Q1) is localized to the subject, checking whether a glass contains liquid (Q2) requires attending to less salient yet relevant regions. These findings underscore that pruning non-subject tokens risks omitting critical visual cues. Thus, a mechanism that reuses or preserves information from these tokens is essential for maintaining both computational efficiency and robust task performance across diverse multimodal scenarios.

Existing methods fall into two main categories. **(1) Training-free approaches** such as PruMerge [37] and SparseVLM [51], merge or prune tokens based on similarity heuristics, but lose access to original token representations and may struggle with task generalization. **(2) Training-based methods** like

LLaVA-Mini [50] and VoCo-LLaMA [48] introduce extra modules to compress visual information before reduction, which adds training and inference overhead and risks losing fine details.

We argue that an ideal solution should satisfy three criteria: **(i) computational efficiency** below $\mathcal{O}(L^2)$; (ii) reuse of discarded token information with **minimal reliance on hand-crafted heuristics**; and **(iii) structural compatibility** with current MLLMs without adding extra modules.

Recent studies [28, 49] have highlighted a phenomenon we term *visual information migration*: in early LLM layers, visual token information is implicitly transferred to question tokens via self-attention. This inspires a new paradigm: instead of compressing or recovering dropped tokens explicitly, can we leverage this migration to distill essential information early in the network?

In this paper, we propose *ParVTS* (Parallel Vision Token Scheduling), a novel token scheduling framework that enables fast and accurate MLLM inference by deliberately decoupling the processing of different visual token types. Rather than treating all visual tokens uniformly, ParVTS partitions them into *subject* and *non-subject* groups based on their attention weights to the [CLS] token in the vision encoder—where higher attention indicates greater semantic relevance to the main visual focus. This soft saliency-based separation reflects each token's potential contribution to downstream reasoning, and is computed efficiently without additional supervision or model components.

Once grouped, these tokens are routed through parallel LLM pathways in a single forward pass, implemented via batch-wise token scheduling. Each pathway carries its own copy of the question tokens and attends to a distinct subset of visual inputs. During the early transformer layers, the inherent attention dynamics of the model facilitate visual information migration [28,49]—where visual tokens, regardless of type, progressively pass their embedded content into the question tokens. This allows each branch to distill the relevant visual context into its question representation over time. After a fixed number of layers, the two sets of question tokens—now enriched with either subject- or non-subject-related information—are merged. Since each branch has already conveyed the essential portion of its visual semantics through attention, the fused question tokens possess a sufficient understanding of the image to guide subsequent reasoning. We then discard the non-subject visual branch and continue inference solely with the subject tokens and merged question tokens, thus achieving significant computational savings while preserving task-relevant information.

This design enables early-stage information transfer from all visual tokens while eliminating redundant computation in later layers. Notably, ParVTS requires no auxiliary modules, heuristics, or fine-tuning, and integrates seamlessly into existing MLLM architectures.

We summarize our main contributions as follows: (1) We introduce a lightweight, inference-time token scheduling framework that reuses non-subject token information without incurring $\mathcal{O}(L^2)$ complexity. (2) We demonstrate how visual information migration in early transformer layers enables implicit knowledge transfer, allowing us to discard non-subject paths mid-inference with minimal

loss. (3) Experiments on multiple MLLM backbones show that ParVTS prunes up to 88.9% of visual tokens while maintaining performance, achieving up to 1.77× speedup and reducing FLOPs by 70%.

2 Related Work

2.1 Multimodal Large Language Models

MLLMs extend traditional language models by incorporating additional modalities such as vision and audio, excelling at VQA and multimodal reasoning [24,29, 30,53]. A typical MLLM architecture consists of a vision encoder and a language model, using lightweight modules for alignment, such as MLPs, Q-Formers, or Resamplers [2,29,53]. Representative models include LLaVA [29], the BLIP family [24,47], and mini-Gemini-HD [25], which integrate CLIP [36] or ViT [13] with language models like LLaMA [40], GPT [1, 35] or Gemma-3 [39]. These models employ fine-tuning or frozen strategies to enable image-to-text generation and cross-modal alignment. Furthermore, recent advancements have expanded MLLMs to video and audio understanding, as exemplified by Video-LLaVA [27] and VideoPoet [22].

A key challenge in MLLMs is their reliance on encoding images or videos into hundreds or even thousands of visual tokens, which are then concatenated with text tokens and jointly processed by the language model. This approach incurs high computational cost due to the quadratic complexity of self-attention [41]. Moreover, the redundancy and low information density of these visual tokens—particularly in high-resolution or multi-frame inputs, as seen in LLaVA [29] and mini-Gemini-HD [25]—have emerged as significant bottlenecks, significantly affecting inference efficiency.

2.2 Visual Token Compression

The problem of visual token redundancy has been studied in the context of vision transformers (ViTs) [13]. For example, CF-ViT [10] adopts a coarse-to-fine processing strategy, while Evo-ViT [46] introduces an adaptive slow-fast token evolution mechanism to reduce redundant computation and improve inference efficiency. In MLLMs, the computational burden caused by excessive visual tokens is even more pronounced, leading to the development of various visual token compression techniques specifically designed to address this issue. FastV [9] selects the most important tokens based on attention scores, retaining only critical information to reduce processing overhead. PruMerge [37] adaptively prunes and merges tokens by measuring their similarity with class tokens, effectively balancing accuracy and efficiency. SparseVLM [51] utilizes cross-modal attention to identify and retain the most relevant visual tokens based on text input, thereby improving token selection and overall model efficiency. These methods leverage distinct strategies to identify and preserve key visual tokens, significantly enhancing the efficiency of MLLMs while maintaining strong performance.

2.3 Visual Information Migration in MLLMs

As research on the internal mechanisms of MLLMs gains traction, recent studies have explored how visual information propagates through transformer layers within LLMs. VTW [28] demonstrates that visual information rapidly migrates to question tokens via causal self-attention in early layers, after which the visual tokens become largely redundant, allowing for their removal in later layers to enable more efficient inference. HiMAP [49] proposes a staged migration process: in shallow layers, visual tokens inject information into question tokens, while in middle layers, they mainly engage in intra-visual aggregation, suggesting a transition from cross-modal fusion to intra-modal consolidation. Cross-modal Information Flow [52] refines this understanding by identifying two distinct phases of visual-to-text transfer: an initial injection of global visual semantics into question tokens, followed by a more focused transfer of task-relevant regional features. Ultimately, the final predictions rely on the transformed textual representations.

3 Methodology

3.1 Preliminary Observation and Motivation

Modern MLLMs, such as LLaVA [30], typically consist of three core components: a vision encoder, a cross-modal projector, and a pre-trained LLM. The vision encoder (*e.g.*, CLIP ViT-L [36]) extracts visual patch features and maps them into the language embedding space via a projector, yielding visual tokens aligned with text representations. Given a multimodal input, the system encodes the task instruction (*i.e.*, the system prompt), the user query, and visual tokens. These are tokenized into system tokens, textual tokens, and visual tokens, respectively. During autoregressive decoding, previously generated outputs are appended to the input sequence. At the first transformer layer ($i = 1$), the full input is formulated as:

$$X_1^t = \text{concat}(S_1, V_1, T_1, O_1^t), \tag{1}$$

where S_1, V_1, and T_1 represent the embeddings of the system prompt, visual input, and user query, respectively; O_1^t denotes the generated output tokens at time step t, with $O_1^0 = \emptyset$ when $t = 0$. For subsequent layers ($i > 1$), the hidden representations are updated through a stack of N transformer layers as:

$$X_i^t = \text{TransformerLayer}_i(X_{i-1}^t), \quad \text{for } i = 2, \ldots, N. \tag{2}$$

At each decoding step t, the model predicts the next token based on the final hidden representation.

Observation. In real-world MLLM applications, user queries typically require grounding in the image content represented by V_1. Notably, the number of visual tokens—especially for high-resolution images—often dominates the input sequence length, introducing substantial computational overhead. Empirical observations (see Fig. 1) reveal that most queries focus on salient foreground entities

or regions. This suggests that a subset of visual tokens, denoted $V_1^{\text{sub}} \subseteq V_1$, is often sufficient for correctly answering the majority of vision-grounded questions. This motivates an efficiency-oriented reformulation of the input as:

$$X_1^t = \text{concat}(S_1, V_1^{\text{sub}}, T_1, O_1^t), \quad \text{where } V_1^{\text{sub}} \subseteq V_1. \tag{3}$$

To construct V_1^{sub}, we select the top-k visual tokens with the highest attention weights from the [CLS] token in the vision encoder. This strategy is inspired by prior findings [10, 46], showing that attention weights to the [CLS] token effectively capture foreground or salient regions in the image, providing a lightweight and supervision-free proxy for visual importance.

Unlike previous works that leverage [CLS]-guided saliency for unimodal vision tasks [16,42], we integrate this principle into a saliency-adaptive token scheduling framework for efficient multimodal inference. This enables us to identify subject-centric tokens without additional training, while maintaining compatibility with off-the-shelf MLLMs.

Motivation. While this token selection strategy can greatly reduce inference cost, it risks excluding non-subject tokens $V_1^{\text{non}} = V_1 \setminus V_1^{\text{sub}}$, which may carry crucial contextual cues—such as text in the background, transparent objects, or spatial configurations. Fig. 1 shows that approximately 19%˜27% of real-world visual queries rely on such peripheral information.

This raises a central question: *Can we achieve the efficiency of pruning V_1^{non}, while still benefiting from its semantic contributions?*

To address this, we draw inspiration from the phenomenon of *visual information migration* [28, 49], wherein early transformer layers enable semantic transfer from visual tokens to the question tokens via self-attention. This mechanism allows the model to briefly access all visual inputs—both subject and non-subject—and progressively distill their information into a compact question representation. We build on this insight to design a strategy that jointly optimizes computational efficiency and information retention, as detailed below.

3.2 Vision Token Scheduling: When and How Visual Tokens are Used

Given the partitioned visual token sets V_1^{sub} and V_1^{non} described in Sec. 3.1, we now consider a central question: ***When and how should each group of tokens be involved in inference?*** Intuitively, both token types carry complementary visual information—subject tokens reflect salient entities, while non-subject tokens encode contextual or background cues. Efficient utilization requires a scheduling strategy that enables both groups to contribute meaningfully without incurring full attention overhead.

To this end, our vision token scheduling temporally separates subject and non-subject tokens across transformer layers. Specifically, we leverage the phenomenon of *visual information migration*, where visual semantics are transferred into question tokens via self-attention in early layers. We explore two sequential scheduling strategies: (1) Subject-First Scheduling; and (2) Non-Subject-First Scheduling.

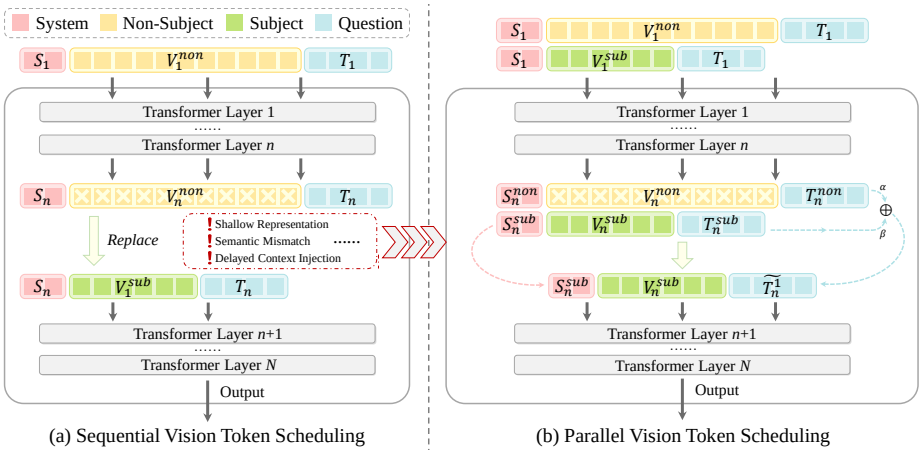

Fig. 2: The framework of ParVTS. (a) Sequential vision token scheduling (taking non-subject-first as an example) injects token groups at different transformer layers, leading to a series of issues. (b) Our parallel scheduling enables both token groups to participate in early layers simultaneously, ensuring sufficient information migration and consistent representation with low inference cost.

Strategy A: Subject-First Scheduling. A straightforward design is to first use subject tokens V_1^{sub} to inject object-centric semantics into question stream, and later introduce non-subject tokens V_1^{non} to provide complementary context.
Stage 1: Early Subject Integration. During the initial n transformer layers, only the subject tokens are included. Through causal attention, question tokens attend to these subject tokens, progressively absorbing salient visual information into their representations:

$$X_n^1 = \text{Transformer}_{1:n}(\text{concat}(S_1, V_1^{\text{sub}}, T_1)), \tag{4}$$

where S_1 and T_1 represent the system and question tokens, respectively.
Stage 2: Late Context Injection. At layer n, we remove the subject tokens and insert the original embeddings of the non-subject tokens—those not previously involved—to continue inference:

$$\tilde{X}_n^1 = \text{ReplaceVision}(X_n^1, V_n^{\text{sub}} \to V_1^{\text{non}}), \tag{5}$$

where ReplaceVision$(\cdot)$ replaces only the visual token slots while preserving ordering and retaining the hidden states of the non-visual tokens.

This allows non-subject tokens to interact with refined question tokens:

$$X_{n+1}^1 = \text{Transformer}_{n+1}(\tilde{X}_n^1). \tag{6}$$

In this stage, attention enables the injection of contextual or peripheral information from V_1^{non} into the question representation.
Efficiency and Limitations. This temporal scheduling reduces overall complexity: early layers operate on $|V_1^{\text{sub}}|$ visual tokens, and later layers on $|V_1^{\text{non}}|$,

both subsets being much smaller than the full token length L. The attention cost drops from $\mathcal{O}(L^2)$ to $\mathcal{O}(L_s^2)$ and $\mathcal{O}((L-L_s)^2)$ in the two respective stages.

While intuitive, relying on a single merge depth n to transition from subject to non-subject tokens inherently induces two trade-offs:

- **Insufficient Subject Transfer at Shallow Merge Depths.** A smaller merge depth n restricts the semantic migration capacity of subject tokens. Since over 70% of questions are subject-oriented (Sec. 3.1), limiting their exposure to only the earliest layers impairs the model's ability to integrate critical object-level information into the question token representations.
- **Delayed Context Injection and Limited Efficiency Gains.** Conversely, a larger n delays the introduction of non-subject tokens, which carry crucial context, thereby diminishing their potential contribution. In addition, under high pruning rates, retaining numerous active non-subject tokens becomes inefficient, as little information is transferred in the later layers.

Strategy B: Non-Subject-First Scheduling. Alternatively, the order can be reversed by injecting V_1^{non} in early layers and switching to V_1^{sub} later, as shown in Fig. 2 (a). This variant eases tension in n selection: since less than 30% of questions rely on non-subject information, even a shallow n allows sufficient migration. Moreover, an earlier switch enables more aggressive pruning of visual tokens in later layers, leading to improved efficiency.

Yet, this strategy inherits two fundamental problems:

- **Shallow representation.** Subject tokens bypass early transformer layers and thus miss hierarchical refinement, resulting in weakly encoded representations.
- **Semantic mismatch.** When injected at layer n, subject tokens are processed in isolation from question tokens that already reside in a higher-level latent space, thereby resulting in semantic misalignment and reduced attention effectiveness.

To overcome these limitations, we propose a parallel execution scheme in the next section that enables both token groups to participate from the beginning, while still maintaining low inference cost.

3.3 Parallel Path Execution for Vision Token Groups

To overcome the representational limitations in our vision token scheduling, we propose a ***parallel execution*** strategy that enables both subject and non-subject tokens to participate in the early transformer layers simultaneously. This ensures comprehensive visual information migration into the question tokens while avoiding the semantic mismatch caused by delayed token injection.

A naive solution would be to process both token groups sequentially, passing each through the same early layers. However, this doubles the compute cost and negates the benefits of scheduling. Instead, we adopt a *batch-parallel execution* design: both visual token groups are processed independently in the same forward pass by concatenating their input sequences along the batch dimension.

As illustrated in Fig. 2(b), we construct two parallel input streams for the first n transformer layers:

$$X_n^{\text{non}} = \text{Transformer}_{1:n}([S_1, V_1^{\text{non}}, T_1]), \tag{7}$$

$$X_n^{\text{sub}} = \text{Transformer}_{1:n}([S_1, V_1^{\text{sub}}, T_1]), \tag{8}$$

where both sequences share the same system and question tokens but differ in visual content.

Let the outputs be decomposed as:

$$X_n^{\text{non}} = [S_n^{\text{non}}, V_n^{\text{non}}, T_n^{\text{non}}], \quad X_n^{\text{sub}} = [S_n^{\text{sub}}, V_n^{\text{sub}}, T_n^{\text{sub}}]. \tag{9}$$

We then combine the question token representations using weighted averaging:

$$\tilde{T}_n^1 = \alpha \cdot T_n^{\text{non}} + \beta \cdot T_n^{\text{sub}}, \tag{10}$$

yielding the unified state:

$$\tilde{X}_n^1 = [S_n^{\text{sub}}, V_n^{\text{sub}}, \tilde{T}_n^1]. \tag{11}$$

Note that $S_n^{\text{non}} = S_n^{\text{sub}}$ as the system tokens are separately processed in both paths but yield identical representations due to the same inputs and model weights. Crucially, to achieve an efficient implementation without duplicating GPU memory for the first n layers, we adopt an optimized engineering approach: we modify the attention mask in the first n layers such that subject and non-subject tokens are mutually invisible, allowing both to flow their information exclusively into the question tokens. After layer n, all non-subject tokens are excluded, and there is no need to store KV caches for them at any layer. Only the subject visual tokens are retained for subsequent inference, which includes the remainder of the prefilling stage and the entire decoding phase, and they continue from layer $n + 1$ following Eq. (6).

Advantages. This design offers three key benefits:

- **Computational efficiency.** Parallel execution avoids duplicated passes while maintaining the $\mathcal{O}(L_s^2)$ complexity in later layers.
- **Representation consistency.** Both visual token groups are refined through early transformer layers, avoiding shallow representations and promoting alignment in feature representations.
- **Information completeness.** The model benefits from early-stage migration of both subject and non-subject information into the question tokens, preserving essential semantics.

4 Experiments

4.1 Experimental Setup

We verify our ParVTS using LLaVA-1.5 [30], LLaVA-Next [29], InternVL3.5 [43], Qwen3-VL [5], and Video-LLaVA [27]. Comparisons are conducted across diverse

Table 1: Comparative experiments on **LLaVA-1.5-7B** and **LLaVA-Next-7B** across multimodal benchmarks. **Bold** indicates the best result.

Methods	GQA	MMB	MME	POPE	SQA	VQA^{v2}	VizWiz	OCRBench	Avg.↑
LLaVA-1.5-7B [30]	*All Tokens (576, 100.00%)*								
Vanilla	61.9	64.1	1866.1	85.9	69.6	78.5	54.3	313.0	100.00 %
	Token Reduction (64, −88.89%)								
FastV$_{ECCV24}$ [9]	49.9	54.5	1466.7	70.8	**69.0**	62.0	54.2	172.0	78.64%
SparseVLM$_{ICML25}$ [51]	52.7	56.2	1505.0	75.1	62.2	68.2	50.1	180.0	78.64%
SAINT$_{25.03}$ [20]	55.5	58.0	1604.2	**84.5**	67.9	67.7	**55.4**	240.0	85.69%
HiRED$_{AAAI25}$ [4]	54.6	60.2	1599.0	73.6	68.2	69.7	50.2	191.0	83.13%
PruneSID$_{ICLR26}$ [14]	**57.1**	58.8	1733.0	83.8	67.8	**73.7**	54.7	274.0	92.75%
ParVTS (Ours)	55.2	**61.3**	**1739.4**	76.9	68.9	70.7	55.1	**282.0**	**93.01%**
LLaVA-Next-7B [29]	*All Tokens (2880, 100.00%)*								
Vanilla	62.5	66.2	1875.1	87.8	68.0	79.5	59.7	510.0	100.00 %
	Token Reduction (320, −88.89%)								
FastV$_{ECCV24}$ [9]	55.9	61.6	1661.0	71.7	62.8	71.9	53.1	374.0	85.87%
SparseVLM$_{ICML25}$ [51]	56.1	60.6	1533.0	82.4	66.1	71.5	52.0	270.0	78.03%
HiRED$_{AAAI25}$ [4]	59.3	**64.2**	1690.0	83.3	66.7	75.7	54.2	**404.0**	88.91%
GlobalCom$^2{}_{AAAI26}$ [31]	57.1	61.8	1698.0	83.8	**67.4**	**76.7**	54.6	375.0	88.09%
PruneSID$_{ICLR26}$ [14]	60.5	63.0	1754.0	83.1	67.3	76.6	58.1	317.0	88.28%
ParVTS (Ours)	**61.2**	63.6	**1761.6**	**84.8**	67.1	76.0	**58.5**	329.0	**89.07%**

Table 2: Comparative results on **Qwen3-VL-2B**. **Bold** indicates the best result.

Method	GQA	MMB	MME	POPE	SQA	OCRBench	VizWiz	Avg. ↑
Qwen3-VL-2B [5]	*All tokens (Dynamic, 100.00%)*							
Vanilla	59.4	76.0	2009.0	90.0	86.5	806.0	67.7	100.00%
	Dynamic Token Reduction (−88.89%)							
FastV$_{ECCV24}$	43.0	64.2	1683.7	80.0	76.9	322.0	63.0	73.02%
ParVTS (Ours)	**58.3**	**74.4**	**2009.6**	**89.6**	**84.9**	**449.0**	**65.3**	**88.62%**

Table 3: Comparative results on **InternVL3.5-2B**. **Bold** indicates the best result.

Method	AI2D	MMB	MME	POPE	MMMU	MathVista	OCRBench	Avg. ↑
InternVL3.5-2B	*All tokens (Dynamic, 100.00%)*							
Vanilla	78.8	76.6	2123.3	87.2	59.0	71.8	836	100.00%
	Dynamic Token Reduction (−88.89%)							
FastV$_{ECCV24}$	63.6	58.4	1991.0	79.1	42.8	**64.0**	439	82.15%
ParVTS (Ours)	**67.6**	**73.7**	**1997.7**	**82.6**	**44.1**	63.1	**483**	**84.37%**

Table 4: Comparative results on **Video-LLaVA-7B**. **Bold** indicates the best result.

Methods	TGIF		MSVD		Avg.↑	
	Accuracy	Score	Accuracy	Score	Accuracy	Score
Video-LLaVA-7B [27]	*All Tokens (2056, 100.00%)*					
Vanilla	47.0	3.4	70.2	3.9	58.6	3.6
	Token Reduction (1032, −49.80%)					
FastV$_{ECCV24}$	**45.2**	**3.1**	**71.0**	**3.9**	**58.1**	**3.5**
ParVTS (Ours)	45.0	**3.1**	70.5	**3.9**	57.8	**3.5**

benchmarks, encompassing: visual question answering (GQA [18], VQAv2 [17], ScienceQA [34], VizWiz-VQA [6], MMB [32], MME [15]), hallucination detection (POPE [26]), video question answering (TGIF-QA [19], MSVD-QA [45]). Additional implementation details, including the specific migration depth n settings for each MLLM backbone, are provided in supplementary material (Sec.8).

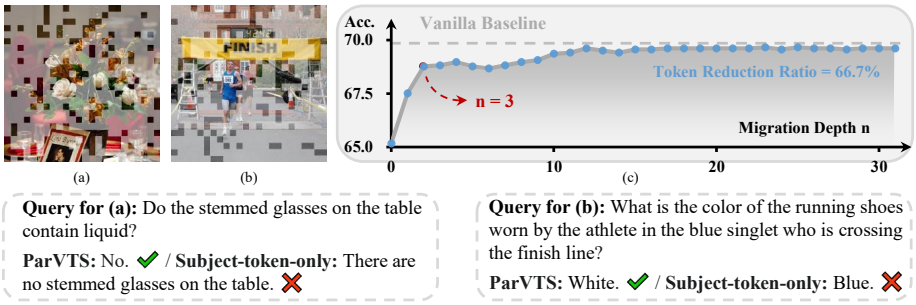

Fig. 3: Visualization of token partitioning and validation of visual information migration. (a-b) Qualitative results on non-subject queries. To visualize the [CLS]-based token partitioning, non-subject tokens are overlaid with a 50% opacity mask. "Subject-token-only" denotes methods that retain only subject tokens without non-subject information migration, which irreversibly loses peripheral visual cues, while ParVTS successfully preserves non-subject semantics. (c) Quantitative layer-wise ablation of migration depth n.

4.2 Main Results

Quantitative Evaluation. Tables 1 $\sim$ 4 present the quantitative results of our ParVTS under a fixed vision token budget across multimodal understanding tasks. In Table 1, on LLaVA-1.5-7B, our ParVTS achieves an average performance of 93.01%, ranking first among all compared methods. This shows that ParVTS well mitigates the performance degradation caused by token reduction, maintaining strong robustness and stability even under aggressive compression.

Also, our ParVTS improves performance across other open-source MLLMs, as shown in Table 2$\sim$ 4. This indicates that ParVTS transfers reliably to diverse model architectures and scales, including InternVL3.5 and Qwen3-VL. Further experiments covering additional vision token budgets, more MLLM architectures (InternVL2.5, Qwen2.5-VL), and diverse model sizes are provided in supplementary material (Sec.9).

Qualitative Examples. To intuitively demonstrate the efficacy of our method, we further provide qualitative comparisons in Fig. 3 (a-b). When answering non-subject queries requiring peripheral details (e.g., verifying the liquid in the glass or the background athlete's shoe color), naive subject-token-only baselines fail entirely due to the irreversible removal of these visual tokens. In contrast, ParVTS consistently outputs accurate responses. This empirically demonstrates that our parallel scheduling retains non-subject semantics in question tokens, ensuring robust visual reasoning for both subject and non-subject queries. Additional cases are detailed in supplementary material (Sec.15).

Downstream Task. We validate the generalization of ParVTS on LISA [23], a fine-grained segmentation task. Results (detailed in supplementary material (Sec.12)) show that ParVTS successfully preserves the original model's segmentation capability across diverse reasoning scenarios, confirming its effectiveness in downstream applications requiring detailed visual understanding.

Table 5: Comparison of token budget, inference speedup, and computational cost under comparable MME scores. **Bold** indicates the best result in each group.

Methods	#Tokens ↓	Time / Sample (ms) ↓	Speedup ↑	TFLOPs ↓	TFLOPs Ratio ↓	MME Score ↑
LLaVA-1.5-7B [30]	576	285.17	1.00×	8.48	100.00%	1866.10
Comparison under similar MME scores (≈1810)						
FastV$_{\text{ECCV24}}$	221	224.94	1.27×	4.63	54.60%	1812.67
SparseVLM$_{\text{ICML25}}$	210	210.19	1.36×	5.85	68.94%	1810.34
ParVTS (Ours)	**161**	**185.34**	**1.54**×	**3.76**	**44.34%**	**1814.03**
Comparison under similar MME scores (≈1750)						
FastV$_{\text{ECCV24}}$	175	188.71	1.51×	4.10	48.35%	1752.60
SparseVLM$_{\text{ICML25}}$	140	193.77	1.47×	5.45	64.23%	1749.23
ParVTS (Ours)	**103**	**173.55**	**1.64**×	**3.16**	**37.26%**	**1774.10**
Comparison under similar MME scores (≈1680)						
FastV$_{\text{ECCV24}}$	135	189.55	1.50×	3.64	42.92%	1676.49
SparseVLM$_{\text{ICML25}}$	75	176.92	1.61×	5.07	59.76%	1687.45
ParVTS (Ours)	**46**	**161.33**	**1.77**×	**2.57**	**30.31%**	**1688.63**

4.3 Cost and Efficiency Analysis

We compare different methods under three configurations with similar MME [15] scores, reporting the number of retained visual tokens, inference latency, and TFLOPs cost at each accuracy level. As shown in Table 5, ParVTS consistently achieves the highest inference accuracy with fewer tokens and lower computational cost. To further evaluate efficiency, we provide fine-grained empirical statistics and analysis across diverse concurrency and response-length settings, reporting latency, GPU peak memory, and TFLOPs for both the prefilling and decoding stages in the supplementary material (Sec.10).

Moreover, we establish a theoretical speedup model, which provides a detailed analysis of how the pruning rate and migration depth influence the acceleration of both the prefilling and decoding stages, as detailed in the supplementary material (Sec.11).

ParVTS also offers better engineering compatibility for efficient deployment. Unlike FastV [9], PruMerge [37], and HiRED [4], which require access to intermediate attention matrices and thus conflict with the computation pattern of Flash-Attention [11, 12], ParVTS remains fully compatible with both Flash-Attention and KV-cache reuse, ensuring seamless integration in real-world deployment.

4.4 Ablation Studies

Layer-wise Analysis of Information Migration. To trace the information transfer of non-subject tokens into question tokens, we conduct a layer-wise ablation on the migration depth n using LLaVA-1.5-7B.

As illustrated in Fig. 3 (c), dropping non-subject tokens prematurely (e.g., at $n = 1$) leads to a sharp performance decline, indicating incomplete cross-token semantic transfer. Conversely, the accuracy rapidly ascends and saturates at $n = 3$, closely matching the vanilla baseline despite a 66.7% token reduction. This layer-wise phenomenon demonstrates that non-subject semantics are effectively transferred to question tokens within the initial transformer layers, rendering them redundant in subsequent layers.

Table 6: Ablation studies of migration depth n and scheduling strategy on LLaVA-1.5-7B under different vision token budgets.

Method	GQA	MMB	MME	POPE	SQA	VizWiz	OCRBench	Avg. ↑
LLaVA-1.5-7B [30]	*All Tokens (576, 100.00%)*							
Vanilla	61.9	64.1	1866.1	85.9	69.6	54.3	313.0	100.00%
	Token Reduction (192, −66.67%)							
$n = 3$	59.7	**63.6**	**1831.9**	**84.6**	68.5	55.3	302.0	**98.04%**
$n = 7$	59.8	63.1	1809.0	84.5	68.2	54.9	299.0	96.96%
$n = 11$	59.9	62.0	1791.8	**84.6**	**68.6**	55.3	**303.0**	96.43%
$n = 15$	**60.5**	61.3	1785.7	82.8	**68.6**	**56.1**	296.0	95.86%
Non-Sub-FS	59.5	63.2	1825.8	83.6	68.0	53.6	293.0	97.28%
Sub-FS	57.1	52.7	1531.5	76.7	65.8	53.2	119.0	77.77%
	Token Reduction (128, −77.78%)							
$n = 3$	58.2	**63.1**	1789.2	82.0	68.6	55.3	297.0	95.96%
$n = 7$	58.3	63.0	**1811.6**	82.2	68.5	55.0	**298.0**	**96.87%**
$n = 11$	58.6	62.8	**1811.6**	**82.5**	68.9	55.3	295.0	96.81%
$n = 15$	**60.0**	62.3	1763.0	82.4	**69.1**	**55.9**	288.0	94.65%
Non-Sub-FS	58.2	62.9	1766.2	81.0	68.3	54.1	291.0	94.69%
Sub-FS	58.8	57.6	1651.5	80.3	66.9	53.7	168.0	84.96%
	Token Reduction (64, −88.89%)							
$n = 3$	55.2	61.3	1739.4	76.9	68.9	55.1	**282.0**	92.91%
$n = 7$	55.4	62.3	1722.7	76.8	68.8	54.9	278.0	92.19%
$n = 11$	55.9	61.7	1725.1	77.6	69.0	55.0	280.0	92.42%
$n = 15$	**58.7**	**62.6**	**1773.5**	**80.3**	**69.2**	**55.3**	274.0	**94.38%**
Non-Sub-FS	55.2	61.2	1671.9	76.0	68.7	54.4	271.0	89.80%
Sub-FS	54.6	61.7	1636.8	79.3	66.5	53.2	210.0	85.97%

Table 7: Ablation studies of fusion weights α and β on LLaVA-1.5-7B.

Method	GQA	MMB	MME	POPE	SQA	VizWiz	OCRBench	Avg. ↑
LLaVA-1.5-7B [30]	*All tokens (576, 100.00%)*							
Vanilla	61.9	64.1	1866.1	85.9	69.6	54.3	313.0	100.00%
	Token Reduction (64, −88.89%)							
α=0.0 β=1.0	53.1	58.8	1723.5	75.2	66.3	52.6	273.0	91.55%
α=0.1 β=0.9	52.9	58.9	1724.3	75.3	66.3	52.7	273.0	91.58%
α=0.3 β=0.7	54.9	60.6	1732.9	76.4	68.1	54.9	276.0	92.39%
α=0.5 β=0.5 (Ours)	**55.2**	**61.3**	**1739.4**	**76.9**	**68.9**	**55.1**	**280.0**	**92.91%**
α=0.7 β=0.3	54.8	60.8	1733.5	76.3	68.7	54.6	276.0	92.43%
α=0.9 β=0.1	55.1	61.1	1738.1	76.6	68.7	54.6	279.0	92.79%
α=1.0 β=0.0	55.1	60.8	1734.7	76.5	68.4	54.6	277.0	92.52%

Migration Depth n. We then investigate how the migration depth n affects model performance. As shown in Table 6, a larger migration depth n is required to maintain performance under more aggressive pruning.

This trend supports the visual information migration principle: under tighter budgets, more non-subject tokens are pruned after layer n, requiring a larger n to ensure sufficient semantic migration before they are discarded. In practice, we set $n = 3$ for LLaVA-1.5-7B and $n = 16$ for LLaVA-Next-7B as default values, to balance accuracy and efficiency. Higher values of n may further improve performance when resources permit. Notably, determining the optimal n requires only a one-time, low-cost sweep on a small validation subset (typically within 10 minutes). This marginal calibration overhead is negligible and fully amortizable over long-term deployment.

Table 8: Ablation studies of sensitivity of [CLS]-based partition on LLaVA-1.5-7B under different vision token budgets.

Method	GQA	MMB	MME	POPE	SQA	VizWiz	OCRBench	Avg. ↑
LLaVA-1.5-7B [30]	*All Tokens (576, 100.00%)*							
Vanilla	61.9	64.1	1866.1	85.9	69.6	54.3	313.0	100.00%
	Token Reduction (192, −66.67%)							
ParVTS (Ours)	**59.7**	**63.6**	**1831.9**	**84.6**	**68.5**	**55.3**	**302.0**	**98.04%**
Swapping	58.1	60.3	1811.4	81.8	65.3	53.6	296.0	96.48%
Random	31.2	39.9	1754.0	52.5	26.1	33.1	231.0	86.20%
	Token Reduction (128, −77.78%)							
ParVTS (Ours)	**58.2**	**63.1**	**1789.2**	**82.0**	**68.6**	**55.3**	**297.0**	**95.96%**
Swapping	55.8	56.2	1761.1	73.5	63.9	50.9	283.0	93.22%
Random	26.3	24.0	1682.9	42.8	21.4	27.1	214.0	81.06%
	Token Reduction (64, −88.89%)							
ParVTS (Ours)	**55.2**	**61.3**	**1739.4**	**76.9**	**68.9**	**55.1**	**282.0**	**92.91%**
Swapping	51.3	57.7	1702.3	72.3	60.1	47.8	269.0	89.88%
Random	22.9	18.3	1610.2	35.5	14.7	21.2	187.0	75.94%

Scheduling Strategy. We further compare two sequential scheduling variants: Subject-First Scheduling (Sub-FS) and Non-Subject-First Scheduling (Non-Sub-FS). As shown in Table 6, both perform consistently worse than our parallel strategy, under the same migration depth $n = 3$, highlighting the structural limitations of sequential token usage. More detailed results with different switching depths are provided in supplementary material (Sec.13).

Fusion Weights α and β. We ablate the fusion weights α and β of Eq. (10) in Table 7. The results show that setting $\alpha = 0.5$ and $\beta = 0.5$ consistently achieves the best performance across benchmarks. More detailed results under different pruning rates are provided in supplementary material (Sec.14).

Reliability and Sensitivity of [CLS]-based Partition. As visualized in Fig. 3 (a-b) and supplementary material (Sec.15), the [CLS] token attention effectively separates subject and non-subject regions, demonstrating its reliability for foreground localization. To further evaluate the sensitivity of ParVTS to partition quality, we conduct experiments under two perturbation settings: Swapping (randomly exchanging 25 tokens between the subject and non-subject groups) and Random (completely random token grouping). As shown in Table 8, the Swapping setting incurs only a minor performance degradation, demonstrating that our framework exhibits robustness against potential token misclassifications. Meanwhile, the Random setting causes a severe performance drop, validating that explicitly distinguishing subject tokens is necessary.

5 Limitations and Future Work

ParVTS uses attention to the [CLS] token in the vision encoder to separate subject from non-subject tokens. While this lightweight, supervision-free strategy aligns with our training-free design, it may occasionally yield incorrect token groupings for complex images with multiple salient regions or subtle foregrounds, and still face challenges with unusually complex non-object-centric queries. Future work could explore more robust, adaptive token grouping methods to enhance visual information selection. Besides, key hyperparameters (the migration depth n and fusion weights α, β) are empirically set. Auto-tuning them by input or task remains an open direction.

6 Conclusion

We present ParVTS, a training-free vision token scheduling framework that leverages early-layer information migration and parallel execution to recover non-subject semantics, enabling fast and accurate MLLM inference. Experiments across diverse benchmarks and compression levels show that ParVTS consistently maintains performance while substantially reducing inference cost. These results highlight the potential of leveraging intrinsic model behaviors for efficient inference, offering new insights into mechanism-aware multimodal reasoning.

Acknowledgements

This work was supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0122701), 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

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. In: NeurIPS (2022)
3. Alvar, S.R., Singh, G., Akbari, M., Zhang, Y.: Divprune: Diversity-based visual token pruning for large multimodal models. In: CVPR (2025)
4. Arif, K.H.I., Yoon, J., Nikolopoulos, D.S., Vandierendonck, H., John, D., Ji, B.: Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models in resource-constrained environments. In: AAAI (2025)
5. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
6. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., et al.: Vizwiz: nearly real-time answers to visual questions. In: ACM UIST (2010)
7. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your ViT but faster. In: ICLR (2023)
8. Card, S.K.: The psychology of human-computer interaction. Crc Press (2018)
9. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In: ECCV (2024)
10. Chen, M., Lin, M., Li, K., Shen, Y., Wu, Y., Chao, F., Ji, R.: Cf-vit: A general coarse-to-fine method for vision transformer. In: AAAI (2023)

11. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023)
12. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. NeurIPS (2022)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
14. Fang, Z., Lyu, P., Zhang, C., Lu, G., Yu, J., Pei, W.: Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. In: ICLR (2026)
15. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2024)
16. Gaotong, Y., Yi, C., Jian, X.: Sparsity meets similarity: Leveraging long-tail distribution for dynamic optimized token representation in multimodal large language models. arXiv preprint arXiv:2409.01162 (2025)
17. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: CVPR (2017)
18. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR (2019)
19. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In: CVPR (2017)
20. Jeddi, A., Baghbanzadeh, N., Dolatabadi, E., Taati, B.: Similarity-aware token pruning: Your vlm but faster. arXiv preprint arXiv:2503.11549 (2025)
21. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: A diagram is worth a dozen images. In: ECCV (2016)
22. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Schindler, G., Hornung, R., Birodkar, V., Yan, J., Chiu, M.C., et al.: Videopoet: A large language model for zero-shot video generation. In: ICML (2024)
23. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: CVPR (2024)
24. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning (2023)
25. Li, Y., Zhang, Y., Wang, C., Zhong, Z., Chen, Y., Chu, R., Liu, S., Jia, J.: Mini-gemini: Mining the potential of multi-modality vision language models. arXiv preprint arXiv:2403.18814 (2024)
26. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: EMNLP (2023)
27. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. In: EMNLP (2024)
28. Lin, Z., Lin, M., Lin, L., Ji, R.: Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In: AAAI (2025)
29. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: CVPR (2024)
30. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
31. Liu, X., Wang, Z., Han, Y., Wang, Y., Yuan, J., Song, J., Zheng, B., Zhang, L., Huang, S., Chen, H.: Compression with global guidance: towards training-free high-resolution mllms acceleration. arXiv preprint arXiv:2501.05179 (2025)

32. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: ECCV (2024)
33. Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.C., Liu, C.L., Jin, L., Bai, X.: Ocrbench: on the hidden mystery of ocr in large multimodal models. Science China Information Sciences (2024)
34. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: NeurIPS (2022)
35. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. In: NeurIPS (2022)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
37. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In: ICCV (2025)
38. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: CVPR (2019)
39. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al.: Gemma 3 technical report. arXiv preprint arXiv:2503.19786 (2025)
40. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
42. Wang, A., Sun, F., Chen, H., Lin, Z., Han, J., Ding, G.: [cls] token tells everything needed for training-free efficient mllms. arXiv preprint arXiv:2412.05819 (2024)
43. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
44. Wu, S., Fei, H., Li, X., Ji, J., Zhang, H., Chua, T.S., Yan, S.: Towards semantic equivalence of tokenization in multimodal llm. arXiv preprint arXiv:2406.05127 (2024)
45. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: ACM MM (2017)
46. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. In: AAAI (2022)
47. Xue, L., Shu, M., Awadalla, A., Wang, J., Yan, A., Purushwalkam, S., Zhou, H., Prabhu, V., Dai, Y., Ryoo, M.S., et al.: Xgen-mm (blip-3): A family of open large multimodal models. arXiv preprint arXiv:2408.08872 (2024)
48. Ye, X., Gan, Y., Huang, X., Ge, Y., Shan, Y., Tang, Y.: VoCo-LLaMA: Towards vision compression with large language models. In: CVPR (2025)
49. Yin, H., Si, G., Wang, Z.: Unraveling the shift of visual information flow in MLLMs: From phased interaction to efficient inference (2024), https://openreview.net/forum?id=0eRJRbVG95
50. Zhang, S., Fang, Q., Yang, Z., Feng, Y.: Llava-mini: Efficient image and video large multimodal models with one vision token. arXiv preprint arXiv:2501.03895 (2025)

51. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: Sparsevlm: Visual token sparsification for efficient vision-language model inference. In: ICML (2025)
52. Zhang, Z., Yadav, S., Han, F., Shutova, E.: Cross-modal information flow in multimodal large language models. In: CVPR (2025)
53. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)