Why and When Visual Token Pruning Fails? A Study on Relevant Visual Information Shift in MLLMs Decoding

Jiwan Kim[1], Kibum Kim[1], Wonjoong Kim[1],
Byung-Kwan Lee[2], Chanyoung Park[1⋆]

[1]Korea Advanced Institute of Science and Technology (KAIST)
{kim.jiwan, rlqja1107, wjkim, cy.park}@kaist.ac.kr

[2]NVIDIA
byungkwanl@nvidia.com

Abstract. Recently, visual token pruning has been studied to handle the vast number of visual tokens in Multimodal Large Language Models. However, we observe that while existing pruning methods perform reliably on simple visual understanding, they struggle to effectively generalize to complex visual reasoning tasks, a critical gap underexplored in previous studies. Through a systematic analysis, we identify Relevant Visual Information Shift (RVIS) during decoding as the primary failure driver. To address this, we propose **D**ecoding-stage **S**hift-aware **T**oken **P**runing (DSTP), a training-free add-on framework that enables existing pruning methods to align visual tokens with shifting reasoning requirements during the decoding stage. Extensive experiments demonstrate that DSTP significantly mitigates performance degradation of pruning methods in complex reasoning tasks, while consistently yielding performance gains even across visual understanding benchmarks. Furthermore, DSTP demonstrates effectiveness across diverse state-of-the-art architectures, highlighting its generalizability and efficiency with minimal computational overhead. Our source code is available here.

Keywords: Multimodal LLMs · Visual Token Pruning · Visual Reasoning

1 Introduction

Multimodal Large Language Models (MLLMs) [4,12,33] have demonstrated strong capabilities in both **visual understanding** (e.g., visual question answering (VQA)) and **visual reasoning**, including visual-centric math, logical puzzles, and STEM-related tasks.[1] However, these models typically rely on a massive number of visual tokens generated by a vision encoder [40, 58]. Such a large number of visual tokens incur prohibitive computational and memory overhead during LLM processing, posing considerable challenges for the practical and efficient deployment of MLLMs.

To mitigate this, a surge of studies [10, 51, 55, 60] has focused on visual token pruning, aiming to reduce computational overhead by removing redundant

⋆ Corresponding author.

[1] STEM stands for Science, Technology, Engineering and Mathematics.

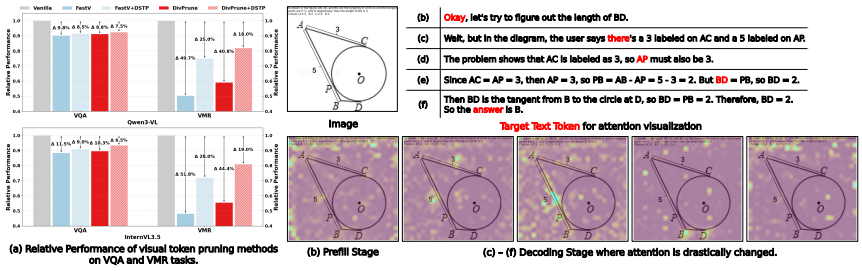

Fig. 1: (a) Performance retention rates of various token pruning methods on Qwen3-VL [5] and InternVL3.5 [48] across VQA and VMR. (b)–(f) Attention heatmaps for a MathVerse [59] sample on Qwen3-VL relative to the text token whose attention shifts drastically, reflecting the reasoning context.[2]Full sentences are provided to offer complete context for each reasoning step.

visual tokens. The core objective of these studies lies in detecting visual tokens that contain **Relevant Visual Information**, which is defined as the essential visual cues required to resolve a given task [60]. Existing pruning methods typically determine the most relevant visual tokens by leveraging internal MLLM attention [10, 51], vision encoder outputs [1, 52], or a hybrid of both [55, 60]. Consequently, by retaining a small number of relevant tokens, these methods significantly accelerate inference while maintaining reliable performance on general visual tasks.

However, most existing pruning methods [1, 10, 60] are primarily evaluated on straightforward visual understanding such as recognizing objects and their attributes using simple VQA benchmarks [17, 34, 43]. Under this paradigm, existing methods have implicitly relied on the expectation that relevant visual information identified during the prefill stage—the initial phase where the model processes the input sequence—would remain sufficient to resolve the given task. This assumption holds for simple **visual understanding** where necessary visual cues are concentrated in narrow regions. However, as the MLLMs evolve toward reasoning-centric models [5, 45, 46], there arises a growing need to address complex **visual reasoning**. Yet, the efficacy of pruning methods remains largely underexplored in these sophisticated step-by-step reasoning tasks, necessitating the exploration of broader visual regions.[3]

To explore this gap, we adopt visual-centric math as a representative task for complex visual reasoning and evaluate prominent pruning methods (i.e., FastV [10] and DivPrune [1]) with state-of-the-arts MLLMs [5, 48] on visual math reasoning (VMR) benchmarks. Furthermore, we also evaluate these methods on simple VQA for comparison.[4] As shown in Fig. 1(a), we observe that while pruning methods effectively preserve performance on VQA, they exhibit precipitous performance drops on VMR. This consistent decline on VMR suggests that existing pruning methods fundamentally struggle to generalize to complex visual reasoning.

To uncover why pruning methods that succeed in VQA fail to generalize to VMR, we compare the visual attention map from the prefill stage (Fig. 1(b)) with

[2] Regarding how the specific text token for the attention heatmap is selected, please refer to Sec. 3.
[3] A detailed comparison between **visual understanding** and **visual reasoning** is in Sec. A.
[4] VQA include SQA [34], GQA [17], and VQA^{T} [43], and VMR include MathVerse [59], WeMath [39], and DynaMath [63]. Fig. 1(a) shows the average performance normalized to vanilla model as 1.0.

those from key decoding steps (Fig. 1(c)–(f)), visualizing how the model's focus transitions across the image throughout the decoding process.[5] Our visualization reveals a dynamic shift as reasoning progresses, demonstrating that the model's visual focus does not remain fixed on the regions identified during the prefill stage; rather, it frequently transitions to entirely different visual areas to align with shifting reasoning requirements. We term this dynamic shift of visual focus during decoding stage the ***Relevant Visual Information Shift (RVIS)***, indicating that the visual evidence required to resolve a task changes over time—a behavior fundamentally distinct from the static visual focus observed in VQA tasks.[5]

In this regard, we hypothesize that the failure of existing pruning methods in VMR stems from the occurrence of RVIS during the decoding stage, which their prefill-stage pruning strategies inherently struggle to address. To validate this hypothesis, we conduct a detailed examination of the attention patterns of MLLMs in Sec. 3 to uncover the root cause of the observed performance degradation. Our diagnostic analysis yields two primary insights: **1)** Visual reasoning exhibits drastic and frequent fluctuations in Relevant Visual Information throughout the decoding process, whereas such shifts are not observed in visual understanding. **2)** This shift, which we term RVIS, serves as the primary driver of pruning failure, with performance declining sharply as its frequency increases during decoding. Consequently, these observations underscore the necessity for an adaptive pruning approach during the decoding stage to effectively address RVIS.

To this end, we introduce **D**ecoding-stage **S**hift-aware **T**oken **P**runing (DSTP), a training-free and simple add-on to existing pruning methods. It adaptively restores previously discarded tokens by detecting RVIS in the model's visual focus during decoding, thereby aligning the visual tokens with the reasoning step. For this, DSTP preserves the tokens discarded by existing pruning methods for potential retrieval during decoding. This mechanism is driven by two integrated modules: the **Relevant Visual Information Shift Detect (RISD)**, which monitors visual attention during decoding to identify RVIS through a threshold-based mechanism; and the **Context-Preserving Visual Token Swap (CPTS)**, which swaps current visual tokens with the newly relevant ones required by the model at that specific decoding step, thereby preventing erroneous generation. This simple *Detect-and-Swap* design enables stable and context-aware reasoning while largely preserving the efficiency gains of conventional pruning methods.

Through extensive experiments, we demonstrate that DSTP significantly mitigates the performance degradation of conventional pruning in visual reasoning tasks, while sustaining or marginally surpassing competitive performance on VQA tasks. Notably, these gains are achieved with minimal computational overhead, largely preserving the efficiency benefits of token pruning.

We summarize our contributions as follows: **1)** We identify, for the first time, **RVIS** as a primary failure driver for existing pruning methods, particularly in reasoning-heavy tasks. **2)** We propose DSTP, a training-free and simple add-on framework that leverages the RISD and CPTS modules to effectively address RVIS. **3)** Extensive experiments demonstrate that DSTP significantly enhances performance in reasoning tasks with minimal computational overhead, proving its effectiveness and robustness in complex reasoning scenarios.

[5] The corresponding attention visualizations for VQA benchmarks are provided in the Sec. B.

2 Preliminary

Multimodal Large Language Models (MLLMs). Following the LLaVA [33] design, MLLMs typically project an image and a text query into a shared token space. Let $X_v \in \mathbb{R}^{N_v \times d}$ and $X_t \in \mathbb{R}^{N_t \times d}$ denote the visual tokens processed through a vision encoder and text tokens, where N_v and N_t represent the number of visual and text tokens, and d is the embedding dimension. The LLM processes the concatenated token sequence $[X_v, X_t] \in \mathbb{R}^{(N_v+N_t)\times d}$ to generate a response.

Visual Token Pruning in MLLMs. The goal of visual token pruning is to select the k most relevant tokens from X_v to reduce computational overhead, where k denotes the token budget (i.e., the number of retained tokens). To quantify the relevance of each visual token, the general attention weight vector $a_{t_l \to v} \in \mathbb{R}^{N_v}$ at any given decoding step l is defined as follows:

$$a_{t_l \to v} = \text{Softmax}\left(q_{t_l} K_v^T / \sqrt{d_m}\right) \tag{1}$$

where $q_{t_l} \in \mathbb{R}^{d_m}$ is the *query* embedding of generated text token at step l, and $K_v \in \mathbb{R}^{N_v \times d_m}$ represents the *key* embeddings of the entire visual token set X_v.

Conventional LLM-based pruning methods [10, 51] typically perform the pruning process as follows: **Prefill Stage ($l = 0$):** In this initial stage, relevant scores are extracted at step $l = 0$, where the query q_{t_0} corresponds to the last token of the text sequence X_t (i.e., the last instruction token). Based on the initial attention vector $a_{t_0 \to v}$, the pruned visual token set $X_v^{\text{Pruned}} \in \mathbb{R}^{k \times d_m}$ under token budget k is then formed by selecting top-k tokens associated with the highest attention weights:

$$X_v^{\text{Pruned}} = X_v[\mathcal{I}], \quad \mathcal{I} = \text{TopK}(a_{t_0 \to v},\, k) \tag{2}$$

where $\text{TopK}(a, k)$ selects the k largest elements in vector a. Subsequently, the reduced sequence $[X_v^{\text{Pruned}}, X_t]$ is fed into the LLM to initiate the decoding process. **Decoding Stage ($l > 0$):** In the decoding phase, the model sequentially generates tokens. Let $G_l = \{g_1, g_2, \ldots, g_l\}$ denote the tokens generated up to step l. At step $l + 1$, the next token g_{l+1} is predicted based on the pruned visual tokens X_v^{Pruned} and the accumulated sequence X_t and G_l:

$$g_{l+1} = \text{LLM}(X_v^{\text{Pruned}}, X_t, G_l) \tag{3}$$

Notably, existing pruning methods [1, 10, 52] permanently discard any visual tokens excluded from X_v^{Pruned} during the prefill stage.

3 Why Does Visual Token Pruning Fail at Visual Reasoning?

This section provides a diagnostic analysis to elucidate the performance disparity between VQA and VMR under existing pruning methods through the lens of the ***Relevant Visual Information Shift (RVIS)***. Our investigation is organized into two stages. First, **Sec. 3.1** identifies the existence of *RVIS* during MLLM inference and characterizes its reasoning-intrinsic nature. Second, **Sec. 3.2** establishes *RVIS* as the primary driver for pruning failure. We demonstrate that performance degradation stems directly from the conflict between these *dynamic* shifts and *static* pruning mechanisms of existing pruning methods, underscoring the critical necessity for an adaptive pruning strategy during the decoding stage.

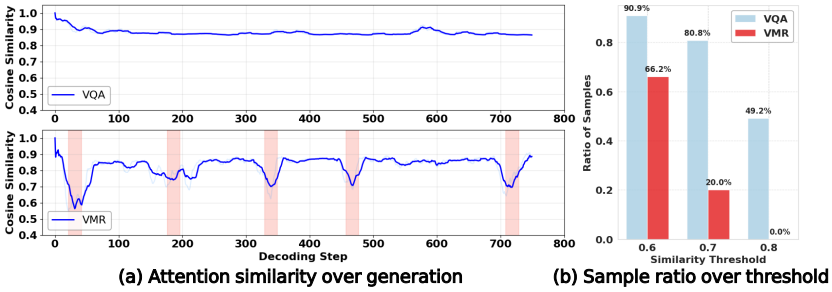

Fig. 2: (a) Cosine similarity of visual attention distributions between the prefill stage ($l = 0$) and each decoding step. (b) Proportion of samples maintaining attention similarity above thresholds throughout the entire decoding process.

3.1 *Relevant Visual Information Shift* as a Hallmark of Visual Reasoning

In this section, we investigate the fundamental attention dynamics that distinguish **visual reasoning** from **visual understanding** in MLLMs, adopting VMR and VQA as representative tasks.[6] Inspired by the qualitative observations in Fig. 1(c)-(f), we hypothesize that attention stability, the degree to which a model maintains its initial visual focus thereafter, serves as a defining hallmark that differentiates these two abilities. To validate this, we perform an attention-based analysis to define ***Relevant Visual Information Shift (RVIS)*** and characterize its behavior. By tracking attention transitions throughout generation, we establish RVIS as the natural phenomenon of visual reasoning, where the model re-focuses on different visual regions as it progresses through sequential logical steps.

***Relevant Visual Information Shift* in MLLMs.** To provide a rigorous examination of the visual focus transitions observed earlier, we investigate the internal visual attention dynamics during the decoding process. Specifically, we quantify the stability of the initial visual focus by calculating the cosine similarity between the visual attention vector at the prefill stage ($a_{t_0 \to v}$) and those at each decoding step l ($a_{t_l \to v}$), denoted as $Sim(a_{t_0 \to v}, a_{t_l \to v}) \in \mathbb{R}^1$.

As illustrated in Fig. 2(a), we observe a stark contrast in stability between these two tasks. For VQA, the model maintains consistently high similarity, suggesting that it continues to attend to the regions prioritized during the prefill stage. In contrast, VMR exhibits sharp declines in similarity, revealing a transition toward different visual areas to satisfy new informational requirements. We formally define this phenomenon—where the model's focus deviates from the initial focus during the decoding—as ***Relevant Visual Information Shift (RVIS)***. Notably, Fig. 1(c)-(f) corresponds exactly to the point where RVIS is observed, confirming that visual reasoning entails a dynamic process where the necessary visual cues transition across successive reasoning steps. Furthermore, we extended our analysis to a dataset-scale to verify the generalizability of RVIS. In Fig. 2(b), we measure how many samples maintain their initial attention focus throughout the entire decoding process across different similarity thresholds. In VMR, samples are significantly less likely to remain anchored to their prefill

[6] We employ SQA [34] and MathVerse [59] as representative benchmarks for VQA and VMR tasks.

attention compared to VQA. This gap validates that in reasoning-intensive tasks, visual relevance is inherently dynamic, necessitating substantial shifts as the model progresses through logical steps.

Reasoning-Intrinsic Nature of RVIS. To validate that RVIS is intrinsically driven by the reasoning process, not merely by the extended generation lengths typical of VMR, we analyze its occurrence frequency across controlled sequence length intervals.[7] As shown in Fig. 3, the frequency of RVIS remains consistently high in VMR regardless of response length. Notably, even the shortest VMR samples (1-512 tokens) exhibit a higher RVIS frequency than the longest VQA samples (2048-4096 tokens). This disparity confirms RVIS as an inherent hallmark of the reasoning process, rather than a mere side effect of generation length. In this process, visual focus dynamically shifts to align with each successive logical step. We provide additional experiments on the reasoning-intrinsic nature of RVIS in Sec. D.

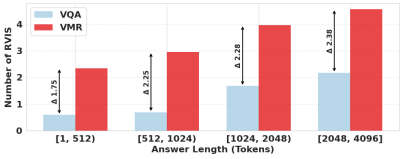

Fig. 3: Average number of RVIS occurrences across various answer lengths.

Discussion. Crucially, RVIS is not a manifestation of model failure but a natural behavior of MLLMs navigating multi-step reasoning. Acknowledging RVIS as a systemic property establishes the necessity for adaptive visual token pruning during decoding when applying pruning methods in the following section.

3.2 How do Relevant Visual Information Shift affect token pruning performance?

Building on our understanding of RVIS, we shift our focus toward its direct impact on existing pruning methods. We argue that the performance discrepancy observed in Fig. 1(a) stems from a fundamental conflict: the *dynamic* informational needs of RVIS versus the *static* pruning mechanisms inherent to existing pruning methods.

Task-Specific Distribution of RVIS Frequency. We first examine the prevalence of RVIS by quantifying occurrences per sample to contrast the divergent RVIS frequency patterns between VQA and VMR.[8] Fig. 4 reveals a stark contrast: while VQA samples are largely static, with the vast majority exhibiting zero or minimal RVIS, VMR samples exhibit two or more shifts during the decoding. This re-confirms that RVIS is fundamentally driven by reasoning intensity. As task complexity increases, the MLLM must transition its visual focus to gather the necessary evidence for each sequential logical step.

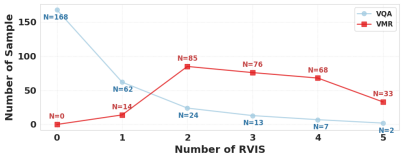

Fig. 4: Distribution of RVIS occurrences for VQA and VMR. N indicates the sample count for each bin.

[7] To identify RVIS, we detect similarity drops below a threshold of *0.7*. To ensure each RVIS event is counted as a single occurrence rather than multiple consecutive steps below the threshold, we utilize the `find_peaks` algorithm on the similarity signal. Further details are provided in Sec. C.

[8] We initially sampled 300 instances per task. After filtering those truncated by the 4,096 `max_generation_len`, we balanced the datasets to match the smaller group, resulting in $N = 276$ samples for each task.

Impact of RVIS on Pruning Success. To establish the direct impact of RVIS when combined with existing pruning methods, we analyze the pruning success rate across varying RVIS frequencies. For this, we define the success rate as the ratio of samples correctly solved after pruning relative to those solved in vanilla model which utilizes the entire visual token set without pruning. As illustrated in Fig. 5, the success rate of pruning methods drops precipitously with increasing RVIS frequency, a robust trend that holds for both tasks. This provides direct evidence that the failure of existing pruning methods is rooted in their inability to adapt to the model's shifting visual focus during decoding. Specifically, ***when*** RVIS occurs—requiring visual information that deviates from that identified during the initial prefill stage—static pruning methods that discard tokens at the prefill become inherently prone to failure. This illustrates exactly ***why*** existing pruning methods struggle with reasoning-heavy tasks like VMR, as observed in Fig. 1(a).

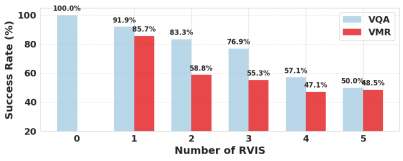

Fig. 5: Success rate of FastV [10] across different RVIS frequencies.

In summary, our analysis identifies the Relevant Visual Information Shift (RVIS) as the primary failure driver that limits the efficacy of existing *static* pruning methods, highlighting the need for an *adaptive* framework that can update the visual tokens as the MLLM's focus transitions during decoding.

4 DSTP: Decoding-stage Shift-aware Token Pruning

Building on our analysis in Sec. 3, we propose **D**ecoding-stage **S**hift-aware **T**oken **P**runing (DSTP), a simple yet effective add-on framework designed to rectify the limitation of existing static pruning methods by adaptively updating visual tokens during decoding. To this end, DSTP integrates **Relevant Visual Information Shift Detect (RISD)** to detect the onset of RVIS throughout the decoding process (Sec. 4.1). This signal triggers **Context-Preserving Visual Token Swap (CPTS)** to retrieve newly relevant visual tokens, guiding MLLM to re-focus the necessary visual cues at those specific decoding steps (Sec. 4.2). The overall framework is shown in Fig. 6.[9]

4.1 Relevant Visual Information Shift Detect (RISD)

Prefill-stage Setup. DSTP follows the original prefill-stage protocol of base pruning methods [10,60], which is illustrated in Fig. 6(a), as its core functionality is designed to operate during decoding. To enable RVIS detection, the RISD module first extracts the initial attention vector $a_{t_0 \to v} \in \mathbb{R}^{N_v}$ during the prefill stage. This vector, which is used to determine the important visual tokens in Eq. (1), serves as the anchor attention to represent the model's initial visual focus. Also, we define the *Reserved Visual Tokens Set* $X_v^{\text{Reserved}} = X_v - X_v^{\text{Pruned}} \in \mathbb{R}^{(N_v-k)\times d_m}$, which consists of the tokens discarded but kept in standby for potential recovery.

Decoding-stage Monitoring. As decoding progresses, the MLLM may require visual cues that deviate from the initial focus captured by the anchor attention

[9] The detailed algorithm for DSTP is in Sec. F.

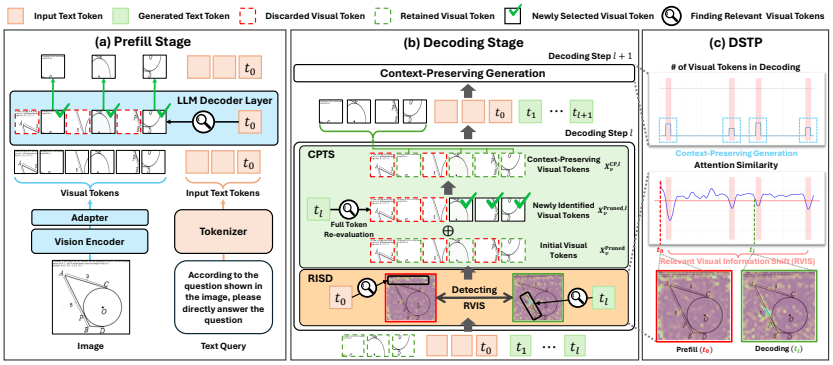

Fig. 6: Overall framework of DSTP. (a) Prefill-stage protocol of base pruning methods. (b) RISD and CPTS modules at Decoding-stage. RISD monitors attention similarity at each decoding step, invoking CPTS for context-preserving token swapping when RVIS is detected. (c) Overall flow of DSTP throughout the entire decoding process.

vector $a_{t_0 \to v}$ at prefill-stage ($l = 0$). Based on our analysis in Sec. 3, we argue that RVIS characterizes these informational needs; therefore, as depicted in the RISD module of Fig. 6(b), we track the current visual focus (t_l) and compare it with the initial visual focus (t_0) at each decoding step to detect its onset. Specifically, we monitor the attention vector $a^{\text{Pruned}}_{t_l \to v}$ at each decoding step l:

$$a^{\text{Pruned}}_{t_l \to v} = \text{Softmax}(q_l (K_v^{\text{Pruned}})^\top / \sqrt{d_m}) \in \mathbb{R}^k \tag{4}$$

where k is the token budget, $q_l \in \mathbb{R}^{d_m}$ is the *query* embedding of generated text token at step l, and $K_v^{\text{Pruned}} \in \mathbb{R}^{k \times d_m}$ denotes the *key* embedding of the visual tokens set X_v^{Pruned} retained by base pruning methods during the prefill stage. To quantify the alignment between the model's current visual focus and the initial anchor, we calculate the cosine similarity $s_l \in \mathbb{R}$ at each decoding step l:

$$s_l = \text{Sim}(a^{\text{Pruned}}_{t_0 \to v}, a^{\text{Pruned}}_{t_l \to v}) = \frac{a^{\text{Pruned}}_{t_0 \to v} \cdot a^{\text{Pruned}}_{t_l \to v}}{|a^{\text{Pruned}}_{t_0 \to v}||a^{\text{Pruned}}_{t_l \to v}|} \tag{5}$$

As illustrated in Fig. 6(c), a decline in s_l indicates an attention shift toward visual regions that are different from the initial focus, signaling the onset of RVIS. We utilize a constant threshold τ to detect RVIS, as it provides a computationally lightweight yet robust trigger, performing comparably to more complex divergence metrics as demonstrated in Sec. 5.3. Once $s_l < \tau$ is detected, the CPTS module is immediately invoked to re-evaluate and retrieve the relevant visual tokens based on the current text query.

4.2 Context-Preserving Visual Token Swap (CPTS)

Dynamic Importance Re-evaluation. To pinpoint the precise visual tokens required for the current decoding step, it is essential to consider the complete set X_v, including those previously discarded by the baseline pruning method during the prefill stage. Consequently, we perform a comprehensive re-evaluation of X_v with respect to the current text query q_l. Specifically, we leverage the attention vector $a^{\text{Pruned}}_{t_l \to v}$, which is internally computed as part of LLM decoding

process. To assess the discarded visual tokens, we additionally compute $a_{t_l \to v}^{\text{Reserved}}$ for X_v^{Reserved} using Eq. (4). By concatenating these weights, we construct the full importance score across the entire visual token set:

$$a_{t_l \to v} = \left[a_{t_l \to v}^{\text{Pruned}}, a_{t_l \to v}^{\text{Reserved}}\right] \in \mathbb{R}^{N_v} \tag{6}$$

This approach enables a full importance scoring across X_v with minimal computational overhead, allowing the selection of a newly relevant top-k visual token set, $X_v^{\text{Pruned},l} \in \mathbb{R}^{k \times d_m}$, based on the updated scores $a_{t_l \to v}$. For instance, let $v_1, \ldots, v_6$ denote the visual tokens in the CPTS module of Fig. 6(b) from left to right. In this case, $X_v^{\text{Pruned},l}$ identifies $\{v_4, v_5, v_6\}$ as newly relevant, which differs from the prefill-stage set X_v^{Pruned} consisting of $\{v_2, v_4, v_6\}$. This ensures that the MLLM retrieves the most appropriate visual tokens in alignment with the current decoding steps.

Context-Preserving Visual Token Swap. Visual tokens provide the essential image context that allows MLLMs to perform a wide range of multimodal tasks. However, naively replacing the initial visual token set X_v^{Pruned} with a newly identified set $X_v^{\text{Pruned},l}$ can hinder the model's ability to maintain a stable understanding of the image, as there is little semantic overlap between the two sets. Such a sudden context shift disrupts the model's internal consistency, leading to erroneous outputs. To address this, we introduce the *Context-Preserving Visual Token Swap*. Rather than a complete replacement, we merge the initial prefill-stage visual tokens X_v^{Pruned} with the newly relevant tokens $X_v^{\text{Pruned},l}$ through a union operation to form a unified (context-preserving) visual token set $X_v^{CP,l}$:

$$X_v^{CP,l} = X_v^{\text{Pruned}} \cup X_v^{\text{Pruned},l} \in \mathbb{R}^{k^* \times d_m} \tag{7}$$

where k^* denotes the size of the union set ($k \leq k^* \leq 2k$). In this scenario, $X_v^{CP,l}$ is $\{v_2, v_4, v_5, v_6\}$, where the newly added v_5 guides the current reasoning step while the remaining tokens sustain the original visual context. To ensure the MLLM effectively incorporates these new tokens, we maintain this temporarily expanded set for a context-preserving duration of L decoding steps as illustrated in Fig. 6(c). After this duration, the initial prefill-stage set X_v^{Pruned} is reused as the visual context to minimize additional re-evaluation overhead. Furthermore, as the model's focus typically aligns with the initial anchor in the absence of RVIS, as shown in Sec. 3, X_v^{Pruned} remains sufficient for the subsequent decoding steps.[10] This transient increase in the number of visual tokens during the context-preserving duration does not incur long-term computational inefficiency, as confirmed by our analysis in Sec. 5.3.

5 Experiment

5.1 Experimental Settings

Datasets. To validate the effectiveness of DSTP in reasoning-heavy scenarios, we categorize our benchmarks into two primary domains. First, we focus on **Visual Reasoning**, which is further subdivided into: (i) *Visual Math Reasoning* (i.e., MathVerse [59], WeMath [39], and DynaMath [63]), requiring rigorous step-by-step mathematical derivation; and (ii) *Puzzle and STEM-related Reasoning*

[10] Detailed analysis of this reversion strategy is provided in Sec. G.

Table 1: Visual Reasoning Performance of **Qwen3-VL-4B** and **InternVL3.5-8B** under different token retention ratios with various pruning methods and DSTP. Values in parentheses are the proportion relative to the upper bound.

Qwen3-VL-4B [5]						
Method	**MathVerse**	**WeMath**	**DynaMath**	**LogicVista**	**MMMU-Pro**	**Acc. (%)**
Vanilla (Full Tokens)	61.29	48.29	66.48	49.22	37.63	100%
Retain 33.3% Tokens (↓ 66.7%)						
FastV (ECCV'24)	32.23 (52.6%)	25.90 (53.6%)	38.76 (58.3%)	30.64 (62.3%)	19.41 (51.6%)	55.68%
w/ DSTP	52.54 (85.7%)	42.19 (87.4%)	51.00 (76.7%)	41.16 (83.6%)	30.52 (81.1%)	82.90%
DivPrune (CVPR'25)	33.90 (55.3%)	29.32 (60.7%)	41.94 (63.1%)	30.64 (62.3%)	13.08 (34.8%)	55.24%
w/ DSTP	50.25 (82.0%)	42.95 (88.9%)	57.70 (86.8%)	42.70 (86.8%)	28.03 (74.5%)	83.80%
VisionZip (CVPR'25)	36.80 (60.0%)	35.57 (73.7%)	43.16 (64.9%)	29.53 (60.0%)	16.36 (43.5%)	60.42%
w/ DSTP	50.76 (82.8%)	43.62 (90.3%)	52.95 (79.6%)	39.22 (79.7%)	27.39 (72.8%)	81.04%
Retain 22.2% Tokens (↓ 77.8%)						
FastV (ECCV'24)	25.88 (42.2%)	21.43 (44.4%)	35.85 (53.9%)	26.96 (54.8%)	11.50 (30.6%)	45.18%
w/ DSTP	42.76 (69.8%)	31.71 (65.7%)	47.31 (71.2%)	39.05 (79.3%)	26.87 (71.4%)	71.48%
DivPrune (CVPR'25)	26.14 (42.6%)	24.18 (50.1%)	38.36 (57.7%)	27.96 (56.8%)	11.21 (29.8%)	47.40%
w/ DSTP	46.19 (75.4%)	40.00 (82.8%)	52.29 (78.7%)	35.55 (72.2%)	21.66 (57.6%)	73.34%
VisionZip (CVPR'25)	25.98 (42.4%)	23.81 (49.3%)	35.90 (54.0%)	25.50 (51.8%)	13.00 (34.5%)	46.40%
w/ DSTP	40.46 (66.0%)	34.19 (70.8%)	47.48 (71.4%)	37.58 (76.4%)	22.25 (59.1%)	68.74%
InternVL3.5-8B [48]						
Method	**MathVerse**	**WeMath**	**DynaMath**	**LogicVista**	**MMMU-Pro**	**Acc. (%)**
Vanilla (Full Tokens)	42.00	43.62	48.26	52.13	39.42	100.00%
Retain 33.3% Tokens (↓ 66.7%)						
FastV (ECCV'24)	21.57 (51.4%)	28.38 (65.1%)	26.71 (55.3%)	33.08 (63.5%)	18.96 (48.1%)	56.66%
w/ DSTP	40.22 (95.8%)	40.38 (92.6%)	45.38 (94.0%)	48.44 (92.9%)	34.27 (86.9%)	92.44%
DivPrune (CVPR'25)	26.59 (63.3%)	32.62 (74.8%)	32.57 (67.5%)	32.78 (62.9%)	19.42 (49.3%)	63.55%
w/ DSTP	39.59 (94.3%)	41.71 (95.6%)	43.22 (89.6%)	47.87 (91.8%)	31.21 (79.2%)	90.09%
VisionZip (CVPR'25)	24.84 (59.1%)	29.14 (66.8%)	30.54 (63.3%)	34.45 (66.1%)	16.71 (42.4%)	59.54%
w/ DSTP	37.56 (89.4%)	41.52 (95.2%)	43.11 (89.3%)	48.76 (93.5%)	28.26 (71.7%)	87.83%
Retain 22.2% Tokens (↓ 77.8%)						
FastV (ECCV'24)	20.06 (47.8%)	26.90 (61.7%)	24.29 (50.3%)	30.20 (57.9%)	11.60 (29.4%)	49.42%
w/ DSTP	38.94 (92.7%)	37.57 (86.1%)	42.95 (89.0%)	46.08 (88.4%)	31.04 (78.7%)	87.00%
DivPrune (CVPR'25)	23.64 (56.3%)	30.81 (70.6%)	29.02 (60.1%)	34.00 (65.2%)	14.10 (35.8%)	57.61%
w/ DSTP	36.54 (87.0%)	38.86 (89.1%)	39.29 (81.4%)	44.74 (85.8%)	28.21 (71.6%)	82.98%
VisionZip (CVPR'25)	20.55 (48.9%)	26.29 (60.3%)	26.34 (54.6%)	31.76 (60.9%)	12.02 (30.5%)	51.04%
w/ DSTP	34.13 (81.3%)	39.52 (90.6%)	40.51 (83.9%)	43.62 (83.7%)	25.85 (65.6%)	81.01%

(i.e., LogicVista [50] and MMMU-Pro [57]), which assess the complex logical and expert-level analytical capabilities. Furthermore, we evaluate standard **Visual Understanding** benchmarks, including GQA [17], VQA^{T} [43], and SQA [34], to provide a comprehensive comparison with existing pruning literature [10, 52, 60].

Baselines. To demonstrate the backbone-agnostic nature of DSTP as a plug-and-play framework, we evaluate its effectiveness by integrating it with representative visual pruning baselines: FastV [10], DivPrune [1], and VisionZip [52]. We present the performance of each baseline with and without our method to highlight the improvements and validate its robustness across diverse pruning strategies. Also, we provide additional generalizability experiments in Sec. H.

Implementation Details. We implement our method on Qwen3-VL-4B [5] and InternVL3.5-8B [48].[11] The detection threshold τ was set to *0.75* and the context preserving duration L was fixed at *20*. Additional implementation details are provided in the Sec. J.

5.2 Main Results

Visual Reasoning Benchmarks. In Tab. 1, we evaluate DSTP when integrated with diverse pruning methods across various visual reasoning benchmarks. Our analysis reveals the following key insights: **1)** Existing static pruning methods

[11] All experiments are conducted on NVIDIA L40S GPUs.

suffer from a severe performance degradation across all reasoning benchmarks, regardless of the MLLM architecture. This confirms that static pruning methods fundamentally struggle to address RVIS. **2)** Integrating DSTP with these methods consistently overcomes such limitations by significantly improving reasoning performance. This result demonstrates that our framework successfully addresses RVIS, ensuring MLLMs re-focus on the necessary visual tokens as informational needs transition during the reasoning process. **3)** While DSTP achieves consistent performance improvements across all evaluated datasets, the most significant gains are observed in benchmarks characterized by high visual complexity such as MathVerse and MMMU-Pro [11, 57].[12] This performance gap demonstrates that DSTP is more effective in reasoning intensive tasks, where complex logical reasoning is essential.

Visual Understanding Benchmarks. We evaluate DSTP on visual understanding benchmarks using Qwen3-VL-4B, as summarized in Tab. 2.[13] Our key observations are as follows: **1)** Static methods exhibit relatively stable performance in visual understanding compared to reasoning tasks. This is because minimal RVIS in visual understanding tasks, as shown in Sec. 3, allows prefill-stage pruning to remain largely effective. Nevertheless, integrating DSTP consistently yields further performance gains. **2)** The performance boost is particularly significant on SQA. Unlike GQA or VQA^T, the relatively higher complexity of SQA triggers more frequent RVIS, which our adaptive framework effectively addresses by responding to RVIS.

Table 2: Visual Understanding Performance of **Qwen3-VL-4B** with 33.3% retention ratio.

Method	SQA	VQA^T	GQA	Acc.
Vanilla	93.42	81.57	61.82	100%
FastV	87.98	74.82	60.04	94.3%
w/ DSTP	91.84	77.95	60.93	97.5%
DivPrune	86.12	71.36	58.07	91.2%
w/ DSTP	91.36	74.11	60.42	95.5%
VisionZip	90.41	77.10	60.68	96.5%
w/ DSTP	92.08	77.20	61.27	97.4%

5.3 In-Depth Analysis

Effect of Components. In Tab. 3, we evaluate the effectiveness of the RISD and CPTS modules against several baselines, using MathVerse (MV) and MMMU-Pro (MP). ***Effect of RISD:*** We compare our fixed-threshold approach against two alternative detection strategies: *Random* (row (c)), which triggers token swapping randomly with the same frequency as RISD, and *Avg* (row (d)), which utilizes a dynamic thresholding mechanism. Specifically, it calculates the running average of all preceding similarity scores and triggers CPTS when the current s_l falls below this cumulative mean. Our RISD outperforms both alternatives, demonstrating that a constant threshold is sufficiently effective for capturing RVIS compared to either arbitrary timing or more complex running averaging. ***Effect of CPTS:*** We evaluate various swap strategies during context-preserving generation. *Hard* selection (row (f)), which discards previous visual tokens; the *Merge* strategy (row (g)), which compresses previous

Table 3: Ablation for RISD and CPTS.

Row	Detect	Swap		Qwen3 VL		InternVL3.5	
		Strategy	Ratio↓	MV	MP	MV	MP
(a)	Full visual tokens			61.29	37.63	42.00	39.42
(b)	FastV (33.3%)			32.23	19.41	21.57	18.96
(c)	Random	CPTS	38.1%	37.69	22.36	31.09	22.18
(d)	Avg	CPTS	38.1%	51.51	27.68	39.08	32.77
(e)	RISD	Full	100%	**53.04**	29.55	**40.96**	34.16
(f)	RISD	Hard	33.3%	45.05	27.26	36.12	30.69
(g)	RISD	Merge	33.3%	47.58	28.38	37.16	32.54
(h)	**RISD**	**CPTS**	38.1%	52.54	**30.52**	40.22	**34.27**

[12] A comprehensive analysis regarding the visually intensive group (MathVerse and MMMU-Pro) and other benchmarks is provided in the Sec. K

[13] For extended experimental results, please refer to Sec. L.

visual tokens into a single token and concatenates it with the newly identified visual tokens; and *Full* swap (row (e)), which serves as an upper bound that allows all visual tokens. We also report the ratio of visual tokens during the context-preserving duration.[14] CPTS effectively matches the performance of the Full swap with only a marginal increase in the average token ratio. In contrast, Hard and Merge strategies exhibit significantly lower accuracy, highlighting their failure to adequately conserve the visual context required for precise generation.

Robustness to RVIS. To validate the effectiveness of DSTP, we analyze success rates across varying RVIS frequencies (Fig. 7) while maintaining consistent settings (Fig. 5). We observe the following: **1)** In both SQA and MathVerse, the performance of the baseline FastV [10] degrades significantly as the frequency of RVIS increases. This decline underscores the inherent fragility of static pruning methods, which fail to adapt when encountering RVIS during decoding. **2)** The application of DSTP yields consistent performance gains across all RVIS frequencies. Notably the performance gap between DSTP and FastV widens as RVIS becomes more frequent. This widening margin validates the robustness of our framework in addressing RVIS and ensuring the model maintains focus on critical visual tokens during complex reasoning tasks.

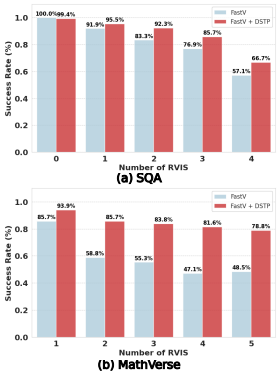

Fig. 7: Success rate of FastV and DSTP across different RVIS frequencies.

Comparative Analysis across TFLOPs. To ensure a rigorous evaluation, we compare its performance against vanilla FastV across varying computational costs. For this, we calculate TFLOPs, which accounts for the total floating-point operations executed during both prefill and decoding stages. For a consistent analysis, all values are normalized to the TFLOPs of vanilla FastV at a 33.3% retention ratio (defined as 1.0).[15] Our observations from Fig. 8 are as follows: **1)** The additional computational cost incurred by DSTP is negligible across all retention ratios. Despite this closely matched cost, DSTP provides a substantial performance gains, suggesting that allocating resources adaptively is far more effective than static pruning. **2)** Remarkably, DSTP at a 33.3% ratio surpasses the performance of vanilla FastV at a much higher 66.6% ratio, while requiring significantly less computational cost. This highlights that providing relevant visual information at the precise decoding steps is far more critical than simply increasing the total number of tokens during the prefill stage. Furthermore, as shown in Fig. 9, DSTP successfully retrieves essential visual context that remains pruned even by the 66.6% static baseline, underscoring the superior adaptability of our framework.

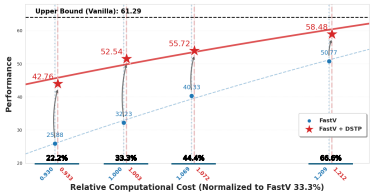

Fig. 8: Performance comparison of FastV versus DSTP across various computational cost on MathVerse [59]

[14] For CPTS, ratio is reported as the average values, reflecting its dynamic token union mechanism.
[15] Refer to the Sec. M for detailed calculation of TFLOPs.

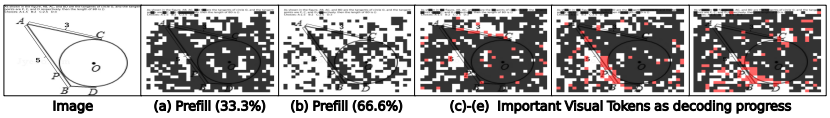

Fig. 9: Visualization of visual token selection. (a) and (b) show FastV results at 33.3% and 66.6% token retention ratios. (c)-(e) illustrate DSTP results at a 33.3% token retention ratio. White and black tokens represent retained and pruned tokens respectively. Red tokens denote those retained by DSTP but pruned even in the 66.6% vanilla FastV.

Hyper-parameter Experiments. We evaluate the sensitivity of L and τ on MathVerse in Fig. 10, noting that $L = 0$ and $\tau = 0$ represent the vanilla FastV baseline. Our observations are as follows: **1)** Performance scales with L, though a sharp decline occurs at lower values. This indicates that a sufficient duration is essential for maintaining logical continuity and stable answer generation. Conversely, performance gains saturate as L increases further, suggesting that a moderate duration is enough to ensure reasoning accuracy without unnecessary overhead. **2)** The threshold τ regulate the tradeoff between RVIS detection with computational cost. At lower values, the model may fail to capture RVIS, resulting in reduced performance. While a higher τ enables more proactive context refocusing, excessively high values lead to computational inefficiency by triggering visual token swapping too frequently. Therefore, a moderate threshold is most effective for providing necessary visual evidence while maintaining operational efficiency.

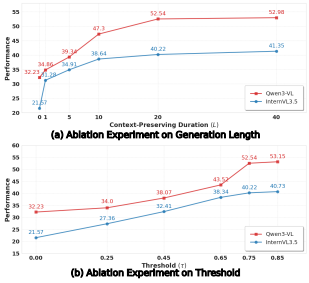

Fig. 10: Hyper-parameter experiments on Generation Length L and threshold τ.

5.4 Efficiency Analysis

While many pruning methods [1, 10, 60] prioritize TFLOPs, which serves as a metric for theoretical computational complexity, recent research [2,56] emphasizes that latency is a more critical measure for realistic deployment. Following [16, 44], we evaluate the computational efficiency of DSTP compared to the Vanilla and FastV models in Tab. 4.[15] Our observations are as follows: **1)** Considering TPS as a direct metric for inference speed, DSTP introduces a minor overhead compared to the FastV baseline due to the dynamic nature of its detect-and-swap mechanism during decoding. However, it still achieves significantly higher efficiency than the Vanilla model while maintaining superior accuracy. This demonstrates that DSTP is an effective framework that maximizes reasoning performance while keeping computational costs substantially lower than the vanilla model. **2)** Notably, DSTP achieves the lowest latency per example (Lat./Ex.) on Qwen3-VL, despite having a lower TPS than FastV. We attribute this efficiency gain to a reduction in the Average Tokens generated. Compared to FastV, DSTP produces fewer tokens while maintaining higher accuracy. This suggests that by providing the most relevant visual context at the appropriate reasoning steps, our framework prevents the model from engaging in redundant or unnecessary reasoning, thereby streamlining the generation process and reducing end-to-end inference time. In

Table 4: Efficiency analysis of DSTP on FastV across Qwen3-VL and InternVL3.5. All efficiency measurements use a 33.3% token retention ratio.

Method	Total Latency ↓	GPU Mem. ↓	Accuracy ↑	Lat./Ex. ↓	TPS ↑	Avg Tokens ↓
Qwen3-VL-4B						
Vanilla	27:51:28	18.6G	61.29	127.27s	19.30	2419.4
FastV	26:07:27	14.6G	32.23	119.35s	**24.99**	2931.7
FastV + DSTP	**22:26:49**	15.1G	52.54	**102.55s**	24.11	**2412.7**
InternVL-3.5-8B						
Vanilla	2:34:53	28.9G	42.00	11.79s	31.55	377.3
FastV	**2:00:03**	25.2G	21.57	**9.14s**	**34.46**	296.8
FastV + DSTP	2:11:12	26.3G	40.22	9.99s	33.79	**227.8**

Sec. N, we provide additional experiments demonstrating the potential of DSTP to further enhance computational efficiency.

6 Related Works

We report a more comprehensive discussion of related works in Sec. O.

Visual Token Pruning for MLLMs. To mitigate the computational bottlenecks of MLLMs, visual token pruning has evolved into two main streams: training-based and training-free. Training-based methods (e.g., LLaVolta [9], ZipR1 [8], VCM [35]) incorporate learnable modules to enforce sparsity but incur significant overhead. Consequently, training-free methods have gained prevalence, utilizing three primary strategies: (1) Vision-Encoder Based: LLaVA-PruMerge [41] and VisionZip [52] utilize encoder-side features like [CLS] attention or CLIP-generated scores. (2) LLM-Based: FastV [10], ZipVL [15], and PDrop [51] prune based on internal LLM attention, while DivPrune [1] treats pruning as a diversity maximization problem. (3) Cross-Modal: SparseVLM [60] and SparseVILA [20] leverage query-aware interactions to guide sparsification. Notably, SparseVILA performs retrieval only at the start of each decoding session per question in multi-turn conversation, without updating the token set within a single answer. In contrast, DSTP updates visual tokens within a single generation, precisely at the steps where visual relevance shifts.

Evaluation of Visual Token Pruning. Recent benchmarks like LLMC+ [36] and UniPruneBench [38] have critically re-examined token pruning metrics, revealing that high pruning ratios often cause severe degradation in detail-sensitive tasks. While existing methods successfully accelerate inference, they frequently discard crucial visual cues required for deep understanding. Our research specifically targets complex visual reasoning, aiming to preserve and effectively retrieve critical information lost in static token pruning, thereby achieving robust reasoning performance without sacrificing efficiency.

7 Conclusion

In this paper, we identify Relevant Visual Information Shift (RVIS) as a critical failure driver in static pruning methods during complex reasoning. To address this, we propose DSTP, a training-free, add-on framework that adaptively aligns visual tokens with the model's shifting focus throughout the decoding stage, significantly mitigating performance degradation with minimal overhead.

We believe this work provides a novel perspective by demonstrating that visual token pruning must move beyond the prefill stage to account for RVIS during

decoding. Our findings underscore that addressing these decoding-stage dynamics is essential for ensuring reasoning integrity while sustaining the efficiency benefits of pruning.

Acknowledgements

This work was supported by National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00335098), National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00406985), and National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT (RS-2022-NR068758).

References

1. Alvar, S.R., Singh, G., Akbari, M., Zhang, Y.: Divprune: Diversity-based visual token pruning for large multimodal models (2025), https://arxiv.org/abs/2503.02175
2. Aminabadi, R.Y., Rajbhandari, S., Zhang, M., Awan, A.A., Li, C., Li, D., Zheng, E., Rasley, J., Smith, S., Ruwase, O., He, Y.: Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale (2022), https://arxiv.org/abs/2207.00032
3. 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 (2024), https://arxiv.org/abs/2408.10945
4. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond (2023), https://arxiv.org/abs/2308.12966
5. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
6. Cai, T., Li, Y., Geng, Z., Peng, H., Lee, J.D., Chen, D., Dao, T.: Medusa: Simple llm inference acceleration framework with multiple decoding heads (2024), https://arxiv.org/abs/2401.10774
7. Cai, Y., Zhang, J., He, H., He, X., Tong, A., Gan, Z., Wang, C., Xue, Z., Liu, Y., Bai, X.: Llava-kd: A framework of distilling multimodal large language models (2025), https://arxiv.org/abs/2410.16236
8. Chen, F., He, Y., Lin, L., Gou, C., Liu, J., Zhuang, B., Wu, Q.: Sparsity forcing: Reinforcing token sparsity of mllms (2025), https://arxiv.org/abs/2504.18579
9. Chen, J., Ye, L., He, J., Wang, Z.Y., Khashabi, D., Yuille, A.: Efficient large multi-modal models via visual context compression (2024), https://arxiv.org/abs/2406.20092
10. 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 (2024), https://arxiv.org/abs/2403.06764
11. Chen, S., Guo, Y., Ye, Y., Huang, S., Hu, W., Li, H., Zhang, M., Chen, J., Guo, S., Peng, N.: Ares: Multimodal adaptive reasoning via difficulty-aware token-level entropy shaping (2025), https://arxiv.org/abs/2510.08457

12. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks (2024), https://arxiv.org/abs/2312.14238
13. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms (2023), https://arxiv.org/abs/2305.14314
14. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., Wu, Y., Ji, R., Shan, C., He, R.: Mme: A comprehensive evaluation benchmark for multimodal large language models (2025), https://arxiv.org/abs/2306.13394
15. He, Y., Chen, F., Liu, J., Shao, W., Zhou, H., Zhang, K., Zhuang, B.: Zipvl: Efficient large vision-language models with dynamic token sparsification (2024), https://arxiv.org/abs/2410.08584
16. Huang, K., Zou, H., Wang, B., Xi, Y., Xie, Z., Wang, H.: Aircache: Activating inter-modal relevancy kv cache compression for efficient large vision-language model inference (2025), https://arxiv.org/abs/2503.23956
17. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering (2019), https://arxiv.org/abs/1902.09506
18. Jiang, D., Zhang, R., Guo, Z., Li, Y., Qi, Y., Chen, X., Wang, L., Jin, J., Guo, C., Yan, S., Zhang, B., Fu, C., Gao, P., Li, H.: Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency (2025), https://arxiv.org/abs/2502.09621
19. Jiang, Z., Chen, J., Zhu, B., Luo, T., Shen, Y., Yang, X.: Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens (2025), https://arxiv.org/abs/2411.16724
20. Khaki, S., Guo, J., Tang, J., Yang, S., Chen, Y., Plataniotis, K.N., Lu, Y., Han, S., Liu, Z.: Sparsevila: Decoupling visual sparsity for efficient vlm inference (2025), https://arxiv.org/abs/2510.17777
21. Kim, B.K., Kim, G., Kim, T.H., Castells, T., Choi, S., Shin, J., Song, H.K.: Shortened llama: Depth pruning for large language models with comparison of retraining methods (2024), https://arxiv.org/abs/2402.02834
22. Kim, J., Kim, K., Seo, S., Park, C.: Compodistill: Attention distillation for compositional reasoning in multimodal llms (2025), https://arxiv.org/abs/2510.12184
23. Lee, B.K., Hachiuma, R., Ro, Y.M., Wang, Y.C.F., Wu, Y.H.: Unified reinforcement and imitation learning for vision-language models (2025), https://arxiv.org/abs/2510.19307
24. Lee, B.K., Hachiuma, R., Ro, Y.M., Wang, Y.C.F., Wu, Y.H.: Genrecal: Generation after recalibration from large to small vision-language models (2026), https://arxiv.org/abs/2506.15681
25. Lee, B.K., Hachiuma, R., Wang, Y.C.F., Ro, Y.M., Wu, Y.H.: Vlsi: Verbalized layers-to-interactions from large to small vision language models (2025), https://arxiv.org/abs/2412.01822
26. Lee, B.K., Wang, Y.C.F., Hachiuma, R.: Masking teacher and reinforcing student for distilling vision-language models (2025), https://arxiv.org/abs/2512.22238
27. Lee, C., Jin, J., Kim, T., Kim, H., Park, E.: Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models (2024), https://arxiv.org/abs/2306.02272
28. Leviathan, Y., Kalman, M., Matias, Y.: Fast inference from transformers via speculative decoding (2023), https://arxiv.org/abs/2211.17192
29. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer (2024), https://arxiv.org/abs/2408.03326

30. Li, Y., Wei, F., Zhang, C., Zhang, H.: Eagle: Speculative sampling requires rethinking feature uncertainty (2025), https://arxiv.org/abs/2401.15077
31. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: Awq: Activation-aware weight quantization for llm compression and acceleration (2024), https://arxiv.org/abs/2306.00978
32. Liu, D., Qin, Z., Wang, H., Yang, Z., Wang, Z., Rong, F., Liu, Q., Hao, Y., Chen, X., Fan, C., Lv, Z., Tu, Z., Chu, D., Li, B., Sui, D.: Pruning via merging: Compressing llms via manifold alignment based layer merging (2025), https://arxiv.org/abs/2406.16330
33. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023), https://arxiv.org/abs/2304.08485
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 (2022), https://arxiv.org/abs/2209.09513
35. Luo, R., Shan, R., Chen, L., Liu, Z., Wang, L., Yang, M., Xia, X.: Vcm: Vision concept modeling based on implicit contrastive learning with vision-language instruction fine-tuning (2025), https://arxiv.org/abs/2504.19627
36. Lv, C., Zhang, B., Yong, Y., Gong, R., Huang, Y., Gu, S., Wu, J., Shi, Y., Guo, J., Wang, W.: Llmc+: Benchmarking vision-language model compression with a plug-and-play toolkit (2025), https://arxiv.org/abs/2508.09981
37. Masry, A., Long, D.X., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning (2022), https://arxiv.org/abs/2203.10244
38. Peng, T., Du, Y., Ji, P., Dong, S., Jiang, K., Ma, M., Tian, Y., Bi, J., Li, Q., Du, W., Xiao, F., Cui, L.: Can visual input be compressed? a visual token compression benchmark for large multimodal models (2025), https://arxiv.org/abs/2511.02650
39. Qiao, R., Tan, Q., Dong, G., Wu, M., Sun, C., Song, X., GongQue, Z., Lei, S., Wei, Z., Zhang, M., Qiao, R., Zhang, Y., Zong, X., Xu, Y., Diao, M., Bao, Z., Li, C., Zhang, H.: We-math: Does your large multimodal model achieve human-like mathematical reasoning? (2024), https://arxiv.org/abs/2407.01284
40. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
41. Shang, Y., Cai, M., Xu, B., Lee, Y.J., Yan, Y.: Llava-prumerge: Adaptive token reduction for efficient large multimodal models (2026), https://arxiv.org/abs/2403.15388
42. Shen, H., Wu, T., Han, Q., Hsieh, Y., Wang, J., Zhang, Y., Cheng, Y., Hao, Z., Ni, Y., Wang, X., Wan, Z., Zhang, K., Xu, W., Xiong, J., Luo, P., Chen, W., Tao, C., Mao, Z., Wong, N.: Phyx: Does your model have the "wits" for physical reasoning? (2025), https://arxiv.org/abs/2505.15929
43. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read (2019), https://arxiv.org/abs/1904.08920
44. Tao, K., Qin, C., You, H., Sui, Y., Wang, H.: Dycoke: Dynamic compression of tokens for fast video large language models (2025), https://arxiv.org/abs/2411.15024
45. Team, K., Du, A., Yin, B., Xing, B., Qu, B., Wang, B., Chen, C., Zhang, C., Du, C., Wei, C., Wang, C., Zhang, D., Du, D., Wang, D., Yuan, E., Lu, E., Li, F., Sung, F., Wei, G., Lai, G., Zhu, H., Ding, H., Hu, H., Yang, H., Zhang, H., Wu, H., Yao, H., Lu, H., Wang, H., Gao, H., Zheng, H., Li, J., Su, J., Wang, J., Deng, J., Qiu, J., Xie, J., Wang, J., Liu, J., Yan, J., Ouyang, K., Chen, L., Sui, L., Yu, L., Dong,

M., Dong, M., Xu, N., Cheng, P., Gu, Q., Zhou, R., Liu, S., Cao, S., Yu, T., Song, T., Bai, T., Song, W., He, W., Huang, W., Xu, W., Yuan, X., Yao, X., Wu, X., Li, X., Zu, X., Zhou, X., Wang, X., Charles, Y., Zhong, Y., Li, Y., Hu, Y., Chen, Y., Wang, Y., Liu, Y., Miao, Y., Qin, Y., Chen, Y., Bao, Y., Wang, Y., Kang, Y., Liu, Y., Dong, Y., Du, Y., Wu, Y., Wang, Y., Yan, Y., Zhou, Z., Li, Z., Jiang, Z., Zhang, Z., Yang, Z., Huang, Z., Huang, Z., Zhao, Z., Chen, Z., Lin, Z.: Kimi-vl technical report (2025), https://arxiv.org/abs/2504.07491

46. Team, V., Hong, W., Yu, W., Gu, X., Wang, G., Gan, G., Tang, H., Cheng, J., Qi, J., Ji, J., Pan, L., Duan, S., Wang, W., Wang, Y., Cheng, Y., He, Z., Su, Z., Yang, Z., Pan, Z., Zeng, A., Wang, B., Chen, B., Shi, B., Pang, C., Zhang, C., Yin, D., Yang, F., Chen, G., Li, H., Zhu, J., Chen, J., Xu, J., Xu, J., Chen, J., Lin, J., Chen, J., Wang, J., Chen, J., Lei, L., Gong, L., Pan, L., Liu, M., Xu, M., Zhang, M., Zheng, Q., Lyu, R., Tu, S., Yang, S., Meng, S., Zhong, S., Huang, S., Zhao, S., Xue, S., Zhang, T., Luo, T., Hao, T., Tong, T., Jia, W., Li, W., Liu, X., Zhang, X., Lyu, X., Zhang, X., Fan, X., Huang, X., Xue, Y., Wang, Y., Wang, Y., Wang, Y., An, Y., Du, Y., Huang, Y., Niu, Y., Shi, Y., Wang, Y., Wang, Y., Yue, Y., Li, Y., Liu, Y., Zhang, Y., Wang, Y., Zhang, Y., Xue, Z., Du, Z., Hou, Z., Wang, Z., Zhang, P., Liu, D., Xu, B., Li, J., Huang, M., Dong, Y., Tang, J.: Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning (2026), https://arxiv.org/abs/2507.01006
47. Wang, K., Pan, J., Shi, W., Lu, Z., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset (2024), https://arxiv.org/abs/2402.14804
48. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., Wang, Z., Chen, Z., Zhang, H., Yang, G., Wang, H., Wei, Q., Yin, J., Li, W., Cui, E., Chen, G., Ding, Z., Tian, C., Wu, Z., Xie, J., Li, Z., Yang, B., Duan, Y., Wang, X., Hou, Z., Hao, H., Zhang, T., Li, S., Zhao, X., Duan, H., Deng, N., Fu, B., He, Y., Wang, Y., He, C., Shi, B., He, J., Xiong, Y., Lv, H., Wu, L., Shao, W., Zhang, K., Deng, H., Qi, B., Ge, J., Guo, Q., Zhang, W., Zhang, S., Cao, M., Lin, J., Tang, K., Gao, J., Huang, H., Gu, Y., Lyu, C., Tang, H., Wang, R., Lv, H., Ouyang, W., Wang, L., Dou, M., Zhu, X., Lu, T., Lin, D., Dai, J., Su, W., Zhou, B., Chen, K., Qiao, Y., Wang, W., Luo, G.: Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency (2025), https://arxiv.org/abs/2508.18265
49. Xiang, K., Li, H., Zhang, T.J., Huang, Y., Liu, Z., Qu, P., He, J., Chen, J., Yuan, Y.J., Han, J., Xu, H., Li, H., Sachan, M., Liang, X.: Seephys: Does seeing help thinking? – benchmarking vision-based physics reasoning (2025), https://arxiv.org/abs/2505.19099
50. Xiao, Y., Sun, E., Liu, T., Wang, W.: Logicvista: Multimodal llm logical reasoning benchmark in visual contexts (2024), https://arxiv.org/abs/2407.04973
51. Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., Lin, D.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction (2025), https://arxiv.org/abs/2410.17247
52. Yang, S., Chen, Y., Tian, Z., Wang, C., Li, J., Yu, B., Jia, J.: Visionzip: Longer is better but not necessary in vision language models (2024), https://arxiv.org/abs/2412.04467
53. Yang, Y., Cao, Z., Zhao, H.: Laco: Large language model pruning via layer collapse (2024), https://arxiv.org/abs/2402.11187
54. Yoon, K., Kim, M., Lee, S., Lee, J., Woo, S., In, Y., Kwon, S.J., Park, C., Lee, D.: Selfjudge: Faster speculative decoding via self-supervised judge verification (2025), https://arxiv.org/abs/2510.02329
55. Yu, H., Li, W., Qu, X., Wang, S., Chen, J., Zhu, J.: Visiontrim: Unified vision token compression for training-free mllm acceleration (2026), https://arxiv.org/abs/2601.22674

56. Yuan, Z., Shang, Y., Zhou, Y., Dong, Z., Zhou, Z., Xue, C., Wu, B., Li, Z., Gu, Q., Lee, Y.J., Yan, Y., Chen, B., Sun, G., Keutzer, K.: Llm inference unveiled: Survey and roofline model insights (2024), https://arxiv.org/abs/2402.16363
57. Yue, X., Zheng, T., Ni, Y., Wang, Y., Zhang, K., Tong, S., Sun, Y., Yu, B., Zhang, G., Sun, H., Su, Y., Chen, W., Neubig, G.: Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark (2025), https://arxiv.org/abs/2409.02813
58. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training (2023), https://arxiv.org/abs/2303.15343
59. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Gao, P., Li, H.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? (2024), https://arxiv.org/abs/2403.14624
60. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., Zhang, S.: Sparsevlm: Visual token sparsification for efficient vision-language model inference (2025), https://arxiv.org/abs/2410.04417
61. Zhu, J., Zhu, Y., Lu, X., Yan, W., Li, D., Liu, K., Fu, X., Zha, Z.J.: Visionselector: End-to-end learnable visual token compression for efficient multimodal llms (2025), https://arxiv.org/abs/2510.16598
62. Zhuang, J., Lu, L., Dai, M., Hu, R., Chen, J., Liu, Q., Hu, H.: St^3: Accelerating multimodal large language model by spatial-temporal visual token trimming (2024), https://arxiv.org/abs/2412.20105
63. Zou, C., Guo, X., Yang, R., Zhang, J., Hu, B., Zhang, H.: Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models (2025), https://arxiv.org/abs/2411.00836