LOOK BACK TO THINK LESS: RETHINKING VISUAL TOKEN PRUNING EFFICIENCY UNDER MULTIMODAL REASONING COST

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token pruning aims to reduce the inference cost of vision-language models (VLMs). Existing methods typically retain a fixed subset of visual tokens during prefill, reducing computation through input compression. However, as VLMs evolve toward stronger reasoners, we observe an overlooked shift in computational burden from prefill to decoding. Discarded visual evidence and fragmented context often trigger repeated verification and multiple answer attempts that substantially inflate generation length. Motivated by this observation, we propose VISTA, a training-free framework that jointly controls visual evidence retention, dynamic access, and reasoning termination. VISTA constructs a compact and diverse visual reserve during prefill, dynamically revisits relevant evidence under a fixed visual budget based on attention trajectories, and uses answer consistency with lightweight periodic probes to terminate reasoning once a stable candidate emerges. We extensively evaluate VISTA on three VLMs across seven multimodal benchmarks covering document understanding, visual reasoning, and medical VQA. Under strict visual budgets, VISTA reduces average generation length by 36.3% while improving average performance by 25.4% over the strongest baseline, highlighting the potential benefits of jointly optimizing visual compression and subsequent decoding for end-to-end inference efficiency.

1 INTRODUCTION

Vision-language models (VLMs) have demonstrated strong capabilities across diverse tasks, including document understanding and visual reasoning (Radford et al., 2021; Liu et al., 2024; Team et al., 2025). Most VLMs encode images into sequences of visual tokens that are jointly processed with text by large language models (LLMs). Preserving fine-grained visual information, however, often entails a substantial number of visual tokens, particularly for high-resolution inputs (He et al., 2025; Huang et al., 2025). The resulting long visual sequences increase computation during prefill and expand the context throughout autoregressive decoding. To mitigate this overhead, visual token pruning has emerged to improve VLM inference efficiency (Chen et al., 2024; Yang et al., 2025).

Existing visual token pruning methods select tokens based on attention (Chen et al., 2024; Xing et al., 2024; Ye et al., 2025) or diversity (Alvar et al., 2025; Yu et al., 2026; Sun et al., 2026), determining a fixed retained set before generation. However, as VLMs develop stronger long-chain reasoning capabilities (Bai et al., 2025; Wang et al., 2025), prefill savings can be offset by additional decoding cost. As presented in Figure 1 and Figure 5, once task-critical visual evidence is removed during prefill, the model may repeatedly verify residual clues and explore alternative interpretations, yet continued generation with still incorrect answers. Recent study (Kim et al., 2026a) suggests that visual attention may drift during reasoning. Thus, tokens selected before generation may not remain sufficient for subsequent decoding. Reasoning-capable VLMs should instead adaptively access additional visual evidence as their information needs evolve during decoding.

Moreover, generation inflation caused by visual pruning is not limited to incorrect predictions. Even when task-critical evidence is preserved, the fragmented visual context can still trigger repeated checking and unstable reasoning. As shown in Figure S3 and Figure 4, the model may eventually reach the correct answer yet still repeatedly verify the same visual details and intermediate con-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

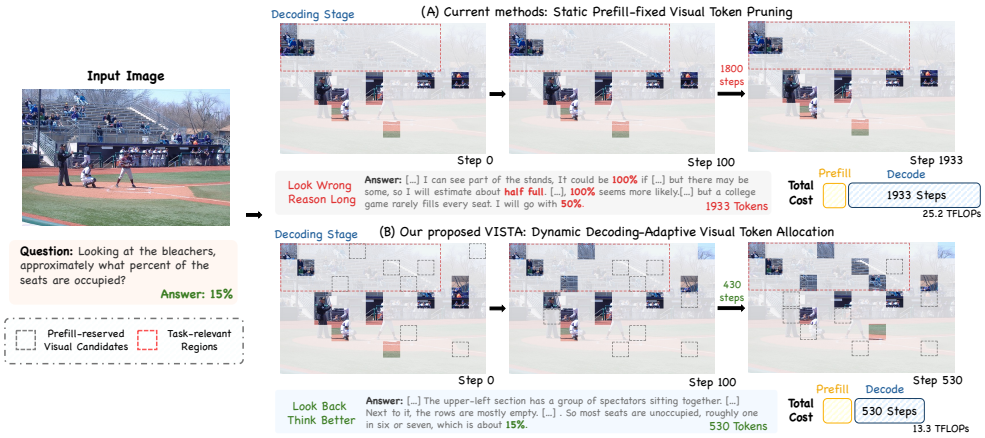

Figure 1: **From prefill-fixed visual pruning to decoding-adaptive VISTA.** *(A)* Current methods fix visual tokens before inference generation, where missed visual information can trigger repeated verification and longer outputs, thereby offsetting prefill savings with increased decoding cost, yet still yielding incorrect answers. *(B)* Our proposed VISTA decouples evidence preservation from step-wise access by maintaining a compact visual reserve, dynamically revisiting relevant evidence as reasoning progresses, thereby mitigating missed evidence and redundant reasoning.

clusions, resulting in substantial redundant reasoning. Some generations even continue until the maximum output length, while the probe results in Figure 4 indicate that the model had already formed a stable tendency toward the correct answer at a much earlier stage. Efficient inference therefore requires coordinating both visual evidence access and reasoning termination.

Motivated by these observations, we propose VISTA, a training-free framework that jointly coordinates visual retention, access, and adaptive reasoning termination. VISTA constructs a compact yet diverse visual reserve during prefill through image scaling and diversity-aware token selection, allowing evidence beyond the initial visible set to be revisited during reasoning. During decoding, VISTA combines smoothed historical attention trajectories with current demands to identify and revisit relevant visual tokens from the reserve at each step under a fixed per-step budget. To reduce redundant reasoning, VISTA monitors the emergence of answer candidates and periodically probes the answer tendency to determine reasoning termination. Together, VISTA recovers relevant visual evidence throughout reasoning while suppressing unnecessary generation once the model reaches a stable answer. Our contributions can be summarized as threefold:

- We revisit visual token pruning for end-to-end inference cost and identify a cost shift in reasoning-intensive VLMs, where incomplete or fragmented visual context after pruning can trigger repeated verification and multiple answer attempts during decoding, leading to reasoning inflation that offsets or even exceeds the computation saved during prefill.
- We introduce VISTA, a flexible training-free framework that reframes visual token pruning as adaptive evidence allocation throughout inference. By decoupling evidence preservation from step-wise access, VISTA allows relevant visual evidence to be revisited as reasoning progresses, while reducing redundant generation once the answer stabilizes.
- We conduct extensive evaluations across three VLMs and seven datasets to systematically examine the trade-off between visual compression and generation efficiency in reasoning-intensive VLMs, where VISTA reduces average generation length by 36.3% and improves performance by 25.4% over the strongest baseline under a strict visual-token budget.

2 METHOD

2.1 OVERALL FRAMEWORK

We propose VISTA, a training-free framework that jointly controls visual evidence retention, revisit, and reasoning termination. Given an image I and a question $Q = (Q_1, \ldots, Q_m)$, VISTA

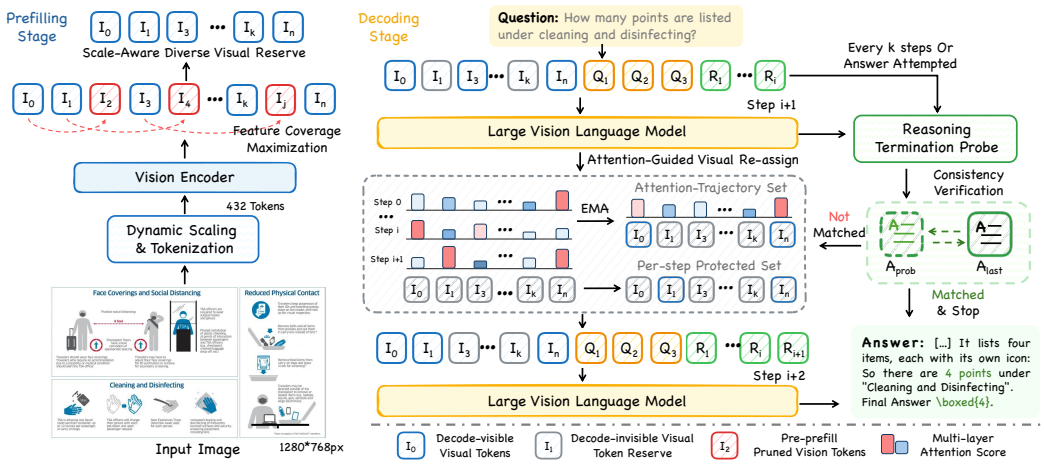

Figure 2: **Overview of proposed VISTA.** We first build a candidate visual reserve by dynamically scaling the image resolution and selecting feature-diverse tokens, which are cached during prefill. During decoding, VISTA reads attention over the reserve and smooths it over time to capture persistent evidence demand. Together with the current step's attention, this historical trajectory guides dynamic updates of the visible subset. Finally, VISTA probes the model's answer tendency when an answer candidate emerges or at periodic checkpoints during long reasoning, terminating generation upon candidate agreement or consistent consecutive probes.

generates a response $R = (R_1, \ldots, R_T)$ under a per-step visual access budget c. As illustrated in Figure 2, VISTA first dynamically scales and encodes I into visual tokens $\{I_i\}_{i=1}^{n}$, then applies feature-coverage maximization to construct a diverse visual reserve $\mathcal{P} \subseteq \{I_i\}_{i=1}^{n}$ of approximately $2c$ tokens with key–value representations cached during prefilling for later steps. During decoding, the model accesses a visible set $\mathcal{E}_\tau \subseteq \mathcal{P}$ with $|\mathcal{E}_\tau| = c$. After generating the response prefix $R_{1:\tau}$, VISTA uses the current attention scores $a_{\tau,i}$ and their exponentially smoothed trajectories $r_{\tau,i}$ over reserve tokens to determine $\mathcal{E}_{\tau+1}$, combining trajectory-guided selection with a per-step protected set that captures immediate visual demands. To determine when to conclude reasoning, VISTA probes the model's answer tendency when an explicit answer candidate emerges or at periodic checkpoints, with probe–candidate consistency triggering response completion. Thus, adaptive visual access and reduced redundant reasoning jointly lower visual and text generation costs.

2.2 Scale-Aware Diverse Visual Reserve

VISTA decouples information retention from step-wise access by a larger visual reserve $\mathcal{P}$ with about $2c$ tokens based on resolution and diversity. N_I denotes the total number of visual tokens for image I before dynamic resizing. Using a frequency-domain approach, we estimate edge spread as $\beta(I) = 1/\nu_{1/2}(I)$, where $\nu_{1/2}(I)$ is the median energy frequency of grayscale gradient spectrum. We compute the calibration value $\beta_0 = \mathrm{Median}_{I_{\mathrm{ref}}}\big(1/\nu_{1/2}(I_{\mathrm{ref}})\big)$ over a set of synthetic sharp-edge images $\{I_{\mathrm{ref}}\}$ composed of random grayscale rectangles. The ratio $\beta(I)/\beta_0$ measures the input image's blur relative to ideal sharp edges and determines the resizing factor $s(I, c)$ as follows:

$$s(I,c) = \Pi_{[s_c,\,1]}\left[\left(\frac{64}{\sqrt{2cN_I}}\,\frac{\beta_0}{\max\{\beta(I),\beta_0\}}\right)^{1/4}\right], s_c = \min\left\{1, \sqrt{\frac{\max\{64, 2c\}}{N_I}}\right\} \quad (1)$$

Here, $\Pi_{[a,b]}(u) = \min\{b, \max\{a, u\}\}$ denotes interval projection. We then obtain N_s visual tokens $\mathcal{V}_s = \{I_i\}_{i=1}^{N_s}$ and construct a diverse visual reserve $\mathcal{P} \subseteq \mathcal{V}_s$ using greedy farthest-point traversal to minimize the feature coverage radius. Specifically, f_i denote normalized final-layer feature of token I_i from visual encoder. We measure feature dissimilarity as $\delta(i,j) = 1 - f_i^\top f_j = \frac{1}{2}\|f_i - f_j\|_2^2$. Let I_{j_t} denote the token selected at step t and $\mathcal{P}_t$ the set of the first t selected tokens. Starting with the

token with the largest nearest-neighbor distance, we iteratively add the candidate farthest from $\mathcal{P}_t$:

$$\begin{aligned} I_{j_1} &\in \arg\max_{u\in\mathcal{V}_s} \min_{v\in\mathcal{V}_s\setminus\{u\}} \delta(u,v), \qquad \mathcal{P}_1 = \{I_{j_1}\}, \\ I_{j_{t+1}} &\in \arg\max_{u\in\mathcal{V}_s\setminus\mathcal{P}_t} \min_{v\in\mathcal{P}_t} \delta(u,v), \qquad \mathcal{P}_{t+1} = \mathcal{P}_t \cup \{I_{j_{t+1}}\} \end{aligned} \tag{2}$$

Thus, dynamic scaling increases visual coverage per processed pixel, while diversity-based selection enriches feature coverage within the limited reserve. We retain the resized-grid positional encodings, prefill with the full reserve, and cache its visual key-value states. The prefill produces R_1, after which decoding starts from the first c selected tokens, $\mathcal{E}_2 = \{I_{j_t}\}_{t=1}^{c}$.

2.3 Attention-Trajectory-Guided Evidence Revisit

As visual needs change during reasoning, VISTA updates the visible set $\mathcal{E}_\tau$ under a per-step visual budget c according to the multi-layer attention, retaining persistent evidence while incorporating newly-step relevant content. After generating response $R_{1:\tau}$, the attention readout at step τ determines the next visible set $\mathcal{E}_{\tau+1}$. We compute attention over visual reserve under multiple LLM layers to integrate information from different depths. Denote the set of attention heads at layer ℓ by $\mathcal{H}_\ell$ and the available text positions by $\mathcal{T}_\tau$. For each position j in $\mathcal{C}_\tau = \mathcal{P}\cup\mathcal{T}_\tau$, we match the current query $q_\tau^{\ell,h}$ with the cached key $k_j^{\ell,h}$, then normalize and average the scores across layers and heads:

$$a_{\tau,i} = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\frac{1}{|\mathcal{H}_\ell|}\sum_{h\in\mathcal{H}_\ell}\exp\left(s_{\tau,i}^{\ell,h} - \log\sum_{j\in\mathcal{C}_\tau}\exp(s_{\tau,j}^{\ell,h})\right), s_{\tau,j}^{\ell,h} = \frac{(q_\tau^{\ell,h})^\top k_j^{\ell,h}}{\sqrt{d}},\ i\in\mathcal{P} \tag{3}$$

Here, d is the dimension of each attention head, and $a_{\tau,i}$ measures the relative relevance of visual token i in the current context, providing a ranking signal for revisiting hidden evidence. Since step-wise attention may fluctuate during generation, relying solely on $a_{\tau,i}$ can cause frequent changes in the visible set. We therefore apply exponential smoothing to form an attention trajectory that captures persistent demand for visual evidence. With τ_0 denoting the initialization step and $r_{\tau_0,i}$ the initial state, $r_{\tau,i}$ represents the model's accumulated attention to token i throughout reasoning:

$$\begin{aligned} r_{\tau,i} &= (1-\alpha)r_{\tau-1,i} + \alpha a_{\tau,i} \\ &= (1-\alpha)^{\tau-\tau_0} r_{\tau_0,i} + \sum_{t=\tau_0+1}^{\tau}\alpha(1-\alpha)^{\tau-t}a_{t,i}, \tau > \tau_0 \end{aligned} \tag{4}$$

Earlier observations thus decay over time, while new ones update the ranking, linking current reasoning needs with recent access history. To respond quickly to sudden attention shifts, we allocate roughly one quarter of the budget to the highest current scores as the protected set $\mathcal{G}_{\tau+1} = \mathrm{Top}_{\max(1,\mathrm{round}(c/4))}(a_\tau;\mathcal{P})$, and fill the remaining slots by trajectory scores:

$$\mathcal{E}_{\tau+1} = \mathcal{G}_{\tau+1}\cup\mathrm{Top}_{c-|\mathcal{G}_{\tau+1}|}(r_\tau;\mathcal{P}\setminus\mathcal{G}_{\tau+1}) \tag{5}$$

Here, $\mathrm{Top}_k(v;A)$ returns the k highest-scoring tokens in A under v. Thus, the protected set captures recent attention, while trajectory scores preserve persistent evidence.

2.4 Consistency-Driven Reasoning Termination

Vision token pruning can fragment visual context, causing restated conclusions. To reduce redundant reasoning, we assess whether reasoning can conclude by checking the consistency between an explicit answer candidate and the model's current answer tendency. We denote $R_{1:\tau}$ as the reasoning text generated so far, with subsequent tokens generated autoregressively according to $p_\theta(R_{\tau+1}\mid I,Q,R_{1:\tau})$. To probe the model's answer tendency at step τ, we temporarily append an answer prompt z to $R_{1:\tau}$ and obtain the resulting next-token distribution.

$$p_\tau^{\mathrm{probe}}(v) = p_\theta(v\mid I,Q,R_{1:\tau}\oplus z), \qquad \hat{v}_\tau = \arg\max_{v\in\mathcal{A}} p_\tau^{\mathrm{probe}}(v), \qquad \kappa_\tau = p_\tau^{\mathrm{probe}}(\hat{v}_\tau) \tag{6}$$

Here, $\oplus$ denotes sequence concatenation, $\mathcal{A}$ is the candidate token set considered by the probe, and $\hat{v}_\tau$ and κ_τ denote the probe output and probability. For those with an explicit answer candidate, we

perform the probe when the candidate is first detected. If the output matches the normalized answer candidate, we treat them as consistent and stop generation; otherwise, generation continues.

For those without potential answers, we probe model's answer tendency at periodic checkpoints. Let τ' denote the preceding probe step. We stop generation only if consecutive probes predict the same candidate and both assign it a probability of at least $\eta = 0.99$, which can be denoted as $\hat{v}_\tau = \hat{v}_{\tau'}$ and $\min\{\kappa_\tau, \kappa_{\tau'}\} \geq \eta$. After probing, we remove the cache increment of z, leaving reasoning context unchanged. Thus, both triggers conclude reasoning when either an explicit answer candidate agrees with the probe output or consecutive probes agree with high confidence.

3 Experiments and Results

3.1 Experiment Setup

Models and Dataset. We evaluate VISTA on three VLMs: Qwen3-VL-4B-Thinking (Qwen3-4B), Qwen3-VL-8B-Thinking (Bai et al., 2025) (Qwen3-8B), and InternVL3.5-4B (Wang et al., 2025). We use 7 multimodal datasets, comprising 6 public benchmarks, DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), ChartQA (Masry et al., 2022), MathVista (Lu et al., 2024), MMMU (Yue et al., 2024), and ScienceQA (Lu et al., 2022), as well as RenalVQA, a private question-answering dataset for kidney pathology. Further details are provided in Appendix A.

Baselines and Metrics. We compare VISTA with 7 methods: FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2024), DivPrune (Alvar et al., 2025), VisionZip (Yang et al., 2025), MMTok (Dong et al., 2026), ZOOPrune (Kim et al., 2026b), and PruneSID (Fang et al., 2026). We follow official evaluation protocol with VLMEvalKit for public benchmarks. We report ANLS for DocVQA and InfoVQA, relaxed accuracy for ChartQA, and dataset-specific accuracy for others, all performance metrics denoted as Acc for simplicity. We also report average generation length (L).

Implementation Details. All methods within each model family use the same prompts, sampling settings, and generation budget. For Qwen3-VL-4B/8B-Thinking, we set the temperature to 1.0, top-p to 0.95, top-k to 20, and the maximum generation length to 16,384 tokens. To avoid the influence of a prompt-prescribed explicit thinking phase on generation length, we close the thinking block before generation, allowing the models to initiate and develop reasoning autonomously as needed. As a complementary setting, we adopt the official Thinking prompt for InternVL3.5-4B and VLMEvalKit configuration to evaluate our method under explicit thinking conditions. We set the temperature to 0.6, top-p to 0.95, top-k to 50, the maximum generation length to 16,384 tokens, and the maximum context length to 24,576 tokens. For VISTA, the attention-trajectory smoothing coefficient is set to $\alpha = 0.5$, and the probe interval to 512 generated tokens when no explicit answer candidate is available. At each decoding step, VISTA uses the same number of visible visual tokens as other pruning baselines to ensure a fair comparison. Further details are provided in Appendix A.

3.2 Reasoning Inflation After Visual Token Pruning

Cost shift. Visual token pruning reduces prefill cost but can increase generation length, shifting computational burden from visual encoding to subsequent decoding. As shown in Figure 3A, when only 10% of visual tokens are retained, decoding becomes the dominant source of computation for

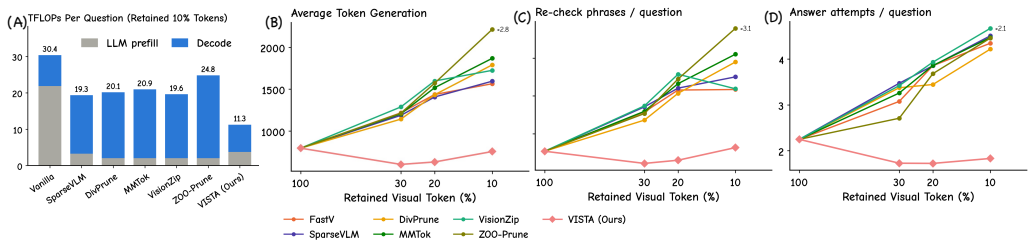

Figure 3: **Cost shift on Qwen3-VL-4B-Thinking.** (A) shows prefill and decoding costs at 10% visual token retention. (B–D) provide generation length, answer attempts, and verification frequency across retention ratios, showing increased reasoning overhead with more aggressive pruning.

existing pruning methods, substantially offsetting the savings during prefill. Figure 3B–D further summarize the average generation length, number of answer attempts, and frequency of verification phrases from six public datasets under different retention ratios. As the retention ratio decreases, all baselines exhibit longer generations, more answer attempts, and more frequent verification, indicating increasingly inflated reasoning after visual compression. Generation trajectories further reveal how this inflation arises. As exemplified in Figure 5A, models repeatedly revisit the remaining visual clues and reconsider candidate answers when relevant evidence is incomplete or fragmented.

Potential Cause of Reasoning Inflation. We close the predefined thinking block for Qwen3 before generation and do not explicitly require a reasoning phase, allowing the model to answer directly while retaining the ability to initiate reasoning when needed. Despite this setup, the thinking-end marker reappears in a substantial fraction of generated responses. As reflected by the bubble sizes in Figure 5B, its occurrence increases from 12% for Vanilla to approximately 20–30% for the pruning baselines, suggesting that visual token pruning more frequently drives the model back into extended reasoning. Even when the model has already formed a clear answer tendency, it often remains more conservative and continues to repeatedly verify and reconsider its prediction, as shown in Figure 4. A possible explanation lies in the interaction between post-training behavior and evidence loss induced by pruning. Qwen3-VL's reasoning-oriented reinforcement learning encourages the model to rely on visual evidence and persist in solving questions that appear answerable (Bai et al., 2025). When pruning removes task-relevant evidence, however, the remaining visual context may still provide partial cues without sufficient information to resolve the answer. This mismatch can cause the model to repeatedly reinterpret residual evidence, verify intermediate hypotheses, and reconsider candidate answers, thus prolonging generation.

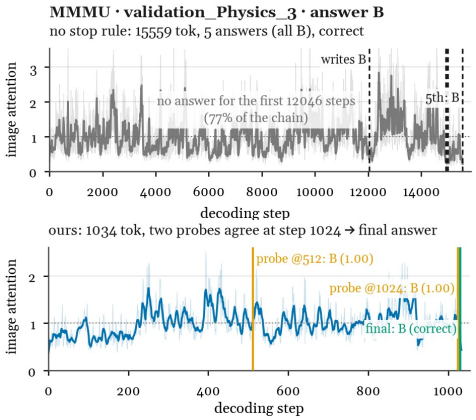

Figure 4: **Over-conservative reasoning case after visual pruning.** Despite a stable answer tendency revealed by the probe, the model continues repeated verification, while our VISTA terminates reasoning once the answer stabilizes.

3.3 Main Results

Results on Qwen3-VL-4B-Thinking. Table 1 summarizes the performance and generation length of different pruning strategies on Qwen3-VL-4B-Thinking across visual token budgets. VISTA achieves the highest average score across all datasets while producing the shortest outputs. At a 10% visual token retention ratio, VISTA maintains an average score of 65.6, exceeding the best baseline by 25.4% while reducing average generation length by 36.3%. Its gains extend across document understanding, chart question answering, multimodal reasoning, and medical question answering, particularly on document and chart benchmarks that require fine-grained visual evidence.

Figure 5A further illustrates these improvements through a representative example. Existing pruning methods retain a static set of visual tokens throughout decoding, making pruned critical regions inaccessible. Without sufficient visual evidence, the model repeatedly rechecks values in the figure and revises its candidate answers, yet still reaches an incorrect conclusion after generating 4,997 tokens and making 9 answer attempts. In contrast, VISTA uses attention trajectories to guide evidence revisiting, bringing relevant tokens from the visual reserve back into view as reasoning evolves. A lightweight probe is then applied to reduce unnecessary verification once the answer has stabilized.

Generalization Across Model Scales and Architectures. We evaluate VISTA on Qwen3-VL-8B-Thinking to examine its generalization across model scales, with results summarized in Table 2 and Table S2. Consistent with Qwen3-VL-4B-Thinking, VISTA achieves the highest average accuracy while maintaining the shortest average generation length under both visual token retention ratios. At 10% visual token retention, its average generation length increases by only 3.8% relative to full-image Vanilla, whereas competing pruning methods increase it by 57.5%–101.2%. These re-

Table 1: **Performance and generation efficiency on Qwen3-VL-4B-Thinking.** L denotes average generated length and Acc denotes accuracy (%). Percentages indicate changes relative to Vanilla.

Method	ChartQA		DocVQA		InfoVQA		MathVista		MMMU		ScienceQA		RenalQA		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens																
Vanilla	499	87.4	183	90.4	516	77.7	2508	75.6	4488	69.6	858	93.2	613	41.8	1381 0.0%	76.5 0.0%
Retain 20% Tokens																
FastV ECCV24	1425	55.5	739	70.4	1556	41.0	2894	46.5	4910	56.1	1023	82.8	897	33.6	1921 +39.1%	55.1 -28.0%
SparseVLM ICML25	1267	56.8	711	69.2	1576	41.6	2793	51.4	5122	57.9	1021	83.4	887	33.7	1911 +38.4%	56.3 -26.4%
DivPrune CVPR25	1379	67.2	644	78.7	1520	50.9	2903	58.4	5288	59.6	1209	87.2	910	36.1	1979 +43.3%	62.6 -18.2%
VisionZip CVPR25	1677	47.1	885	64.2	1843	36.8	2999	46.9	5042	54.2	980	82.5	899	35.1	2046 +48.2%	52.4 -31.5%
MMTok ICLR26	1548	61.3	650	76.7	1689	49.1	3348	53.5	5415	57.3	1081	84.9	889	34.6	2089 +51.3%	59.6 -22.1%
PruneSID ICLR26	1831	60.7	803	74.3	1832	46.6	3261	54.5	5360	58.6	1234	83.9	639	35.5	2137 +54.8%	59.2 -22.7%
ZOO-Prune CVPR26	1463	68.1	790	74.4	1837	48.6	3012	60.9	5164	59.5	1262	85.3	1010	39.1	2077 +50.4%	62.3 -18.6%
VISTA (Ours)	**455**	**79.0**	**141**	**89.7**	**496**	**67.9**	**1583**	**68.0**	**3548**	**65.0**	**706**	**88.4**	**579**	**41.2**	**1073** -22.3%	**71.3** -6.8%
Retain 10% Tokens																
FastV ECCV24	1318	30.3	1230	42.1	1676	23.3	2744	41.2	4392	51.4	894	80.4	988	29.8	1892 +37.0%	42.6 -44.3%
SparseVLM ICML25	1412	28.6	1216	40.6	1687	25.9	2848	44.2	4714	54.2	837	80.2	1006	31.9	1960 +42.0%	43.7 -42.9%
DivPrune CVPR25	2033	49.5	1048	62.9	2034	35.0	3154	48.2	5196	53.7	1088	82.7	1057	34.1	2230 +61.5%	52.3 -31.7%
VisionZip CVPR25	1427	20.4	1595	29.6	1807	20.9	2780	41.7	4250	51.4	**793**	79.4	1056	32.8	1958 +41.8%	39.5 -48.4%
MMTok ICLR26	2062	38.6	1234	57.6	2039	34.0	3310	42.9	5090	54.8	1075	80.2	1018	31.3	2261 +63.8%	48.5 -36.6%
PruneSID ICLR26	2386	33.4	1512	50.7	2397	31.0	3046	46.7	5260	56.0	1159	82.0	810	34.7	2367 +71.4%	47.8 -37.6%
ZOO-Prune CVPR26	2548	44.0	1582	54.6	2621	32.4	3200	50.3	5516	54.0	1110	82.4	1225	32.3	2543 +84.2%	50.0 -34.7%
VISTA (Ours)	**698**	**71.3**	**169**	**87.1**	**629**	**56.7**	**1695**	**58.6**	**3844**	**60.9**	866	**84.7**	**535**	**39.6**	**1205** -12.7%	**65.6** -14.3%

sults show that VISTA scales effectively to larger models, substantially mitigating pruning-induced generation inflation while preserving strong task performance.

We further evaluate on InternVL3.5-4B-Thinking to examine its generalization across model architectures. Following the official explicit thinking prompt provides a complementary setting with explicitly prompted reasoning, as reported in Table 2 and Table S3. VISTA consistently maintains strong accuracy with substantially shorter generation across both retention ratios. At 10% visual token retention, VISTA further achieves 53.8% average accuracy, outperforming the strongest baseline by 9.1 percentage points, while reducing generation length by 59.6%. These results demonstrate that VISTA remains effective across architectures and reasoning configurations.

Efficiency and Stability Analysis. We evaluate end-to-end inference efficiency on Qwen3-VL-4B-Thinking using the same Hugging Face runner, with results summarized in Table S1. Compared with existing baselines, VISTA reduces both the average inference time per question and total LLM

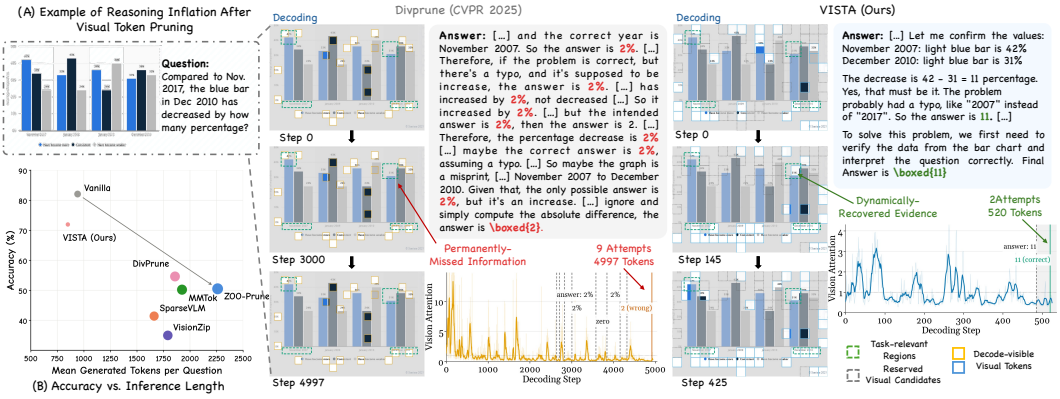

Figure 5: **Inference inflation case and overall performance.** (A) shows a representative case where DivPrune loses key visual evidence after prefill, leading to repeated verification and multiple answer attempts, yet still yielding wrong answer after prolonged generation. VISTA instead dynamically revisits relevant evidence from the visual reserve by historical attention trajectories, reaching correct answer with shorter output. (B) provides accuracy with mean generated tokens on Qwen3-4B under a 10% visual-token retention, where bubble size reflects the proportion of thinking-mode re-entry.

Table 2: **Performance and generation efficiency on Qwen3-VL-8B-Thinking and InternVL3.5-4B.** L denotes average generated length, and Acc denotes accuracy (%). Percentages indicate changes relative to Vanilla. More details are provided in Table S2 and Table S3.

Method	Qwen3-VL-8B-Thinking								InternVL3.5-4B-Thinking							
	ChartQA		InfoVQA		MMMU		Average		ChartQA		InfoVQA		MMMU		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens																
Vanilla	369	87.3	366	79.0	3039	68.3	1258 0.0%	78.2 0.0%	1656	88.2	3190	71.0	6070	69.2	3639 0.0%	76.1 0.0%
Retain 20% Tokens																
FastV ECCV24	1212	50.6	1163	37.7	3823	57.9	2066 +64.2%	48.7 -37.7%	2714	72.3	5446	40.2	6425	64.0	4862 +33.6%	58.8 -22.7%
SparseVLM ICML25	1090	53.8	1063	45.0	3881	59.6	2011 +59.9%	52.8 -32.5%	2805	70.6	5003	40.6	6179	**68.5**	4662 +28.1%	59.9 -21.3%
DivPrune CVPR25	924	65.1	1036	49.6	3903	60.2	1954 +55.4%	58.3 -25.4%	3948	44.6	5164	20.2	6975	55.4	5362 +47.4%	40.1 -47.4%
VisionZip CVPR25	881	68.4	911	55.3	3698	62.9	1830 +45.5%	62.2 -20.5%	4355	36.8	5640	24.1	6654	59.2	5550 +52.5%	40.0 -47.4%
MMTok ICLR26	975	61.2	1000	50.6	3990	59.9	1988 +58.1%	57.2 -26.8%	3477	44.8	4675	22.5	6693	61.2	4948 +36.0%	42.8 -43.7%
ZOO-Prune CVPR26	1017	64.8	1242	47.9	3862	61.6	2040 +62.2%	58.1 -25.7%	4034	49.8	4735	21.4	6853	58.1	5207 +43.1%	43.1 -43.4%
VISTA (Ours)	**351**	**79.8**	**378**	**71.3**	**2550**	**64.2**	**1093** -13.1%	**71.8** -8.2%	**615**	**79.8**	**1110**	**44.9**	**2305**	65.0	**1343** -63.1%	**63.2** -16.9%
Retain 10% Tokens																
FastV ECCV24	1284	29.0	1197	25.2	3462	54.7	1981 +57.5%	36.3 -53.6%	3590	47.4	5493	25.7	6895	61.1	5326 +46.4%	44.7 -41.2%
SparseVLM ICML25	1143	32.8	1227	29.8	3625	56.8	1998 +58.9%	39.8 -49.1%	3810	41.4	5052	27.3	6594	62.9	5152 +41.6%	43.9 -42.4%
DivPrune CVPR25	1409	48.3	1362	38.2	4083	58.9	2285 +81.6%	48.5 -38.0%	3026	28.4	3924	18.4	6796	56.1	4582 +25.9%	34.3 -54.9%
VisionZip CVPR25	1443	43.5	1362	36.6	3811	58.8	2205 +75.3%	46.3 -40.8%	3034	20.3	3889	18.3	6796	53.4	4573 +25.7%	30.7 -59.7%
MMTok ICLR26	1199	37.8	1232	38.8	3743	55.9	2058 +63.6%	44.2 -43.5%	2723	27.7	3759	20.0	6760	55.3	4414 +21.3%	34.3 -54.9%
ZOO-Prune CVPR26	1312	42.0	2031	29.8	4250	56.6	2531 +101.2%	42.8 -45.3%	3455	31.2	4168	19.5	6923	57.3	4849 +33.3%	36.0 -52.7%
VISTA (Ours)	**530**	**66.4**	**438**	**57.7**	**2948**	**61.1**	**1305** +3.8%	**61.7** -21.1%	**928**	**55.4**	**996**	**41.8**	**2483**	**64.2**	**1469** -59.6%	**53.8** -29.3%

computation (TFLOPs) by at least 40%. Figure 3 and Figure S2 further decompose the computational cost across inference stages. By substantially shortening generation, VISTA markedly lowers decoding overhead, whereas the prefill savings of other pruning methods are often offset by prolonged reasoning. To evaluate result stability, we repeat the experiments with three random seeds (0, 1, and 42) and report the corresponding standard deviations in Table S8. VISTA shows lower variation than existing baselines, indicating more consistent performance across random seeds.

3.4 Ablation Study

Generalization of Dynamic VISTA across Baselines. We further evaluate the generality of dynamic VISTA by integrating it with DivPrune, VisionZip, MMTok, and ZOO-Prune, all of which determine the retained visual tokens before decoding. As reported in Table 3 and Table S4, augmenting these methods with VISTA consistently shortens generation while improving prediction accuracy across different pruning baselines and visual token retention ratios. On average, VISTA reduces generation length by approximately 40% while improving accuracy by about 11% over the corresponding original pruning methods. These consistent gains demonstrate that VISTA is complementary to different static token selection strategies, providing a general decoding framework for alleviating pruning-induced reasoning inflation while improving task performance.

Table 3: **Generalization of dynamic VISTA.** We integrate dynamic decoding and consistency-driven reasoning blocks from VISTA with other static baselines at 10% visual token retention on Qwen3-VL-4B-Thinking. L denotes average generated length, and Acc denotes accuracy (%). Percentages indicate changes relative to Vanilla. More details are provided in Table S4.

Method	ChartQA		DocVQA		InfoVQA		MathVista		MMMU		RenalQA		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retained 100% Tokens														
Vanilla	499	87.4	183	90.4	516	77.7	2508	75.6	4488	69.6	613	41.8	1468 0.0%	73.8 0.0%
Retained 10% Tokens														
DivPrune + VISTA	761	67.4	388	79.0	936	50.3	1757	58.5	3837	58.8	540	35.4	1370 -6.7%	58.2 -21.0%
ZOO-Prune + VISTA	920	68.7	492	74.8	1204	49.1	1906	**59.8**	3809	58.9	588	37.0	1487 +1.3%	58.0 -21.3%
VisionZip + VISTA	1010	47.8	577	64.2	1173	38.0	1716	46.4	**3390**	56.8	571	34.8	1406 -4.2%	48.0 -34.9%
MMTok + VISTA	828	61.7	433	77.0	1037	48.6	1979	52.7	3788	57.6	**530**	35.3	1433 -2.4%	55.5 -24.8%
VISTA (Ours)	**698**	**71.3**	**169**	**87.1**	**629**	**56.7**	**1695**	58.6	3844	**60.9**	535	**39.6**	**1262** -14.0%	**62.4** -15.4%

Effect of Functional Blocks. We conduct a comprehensive ablation study to assess the contributions of dynamic scaling (DS), evidence revisit (ER), and reasoning termination (RT) on Qwen3-4B,

as summarized in Tables 4 and S7. Starting from static pruning without any block, DS improves the average accuracy from 51.6% to 54.2%, suggesting that adaptive image scaling helps preserve more visual information under severe token compression. Adding ER further increases the average accuracy to 65.9% and reduces the average generation length from 3598 to 3156 tokens, a 12.3% reduction. This indicates that dynamically revisiting visual evidence according to evolving attention demands can recover missing information while also reducing unnecessary reasoning. Finally, adding RT shortens the generation length further, with a 28.0% reduction, while maintaining comparable accuracy at 66.1%. This confirms that answer-consistency checking can effectively suppress redundant reasoning once the model's prediction has stabilized. With all three blocks enabled, VISTA achieves the best overall performance against static pruning. Together, all functional blocks provide orthogonal yet complementary gains across different stages of inference.

Table 4: **Ablation of VISTA functional blocks on Qwen3-VL-4B-Thinking.** Green squares indicate enabled components. More detailed results are provided in Table S7.

Blocks			ChartQA		MMMU		Average	
DS	ER	RT	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens								
Vanilla			499	87.4	4488	69.6	2494 0.0%	78.5 0.0%
Retain 10% Tokens								
			2033	49.5	5196	53.7	3614 +45.0%	51.6 -34.3%
■			1904	51.9	5292	56.5	3598 +44.3%	54.2 -31.0%
■	■		1088	71.2	5223	60.6	3156 +26.5%	65.9 -16.1%
	■	■	761	67.4	3837	58.8	2299 -7.8%	63.1 -19.6%
■	■	■	**698**	**71.3**	**3844**	**60.9**	**2271** -8.9%	**66.1** -15.8%

Table 5: **Ablation of evidence revisit strategies on Qwen3-VL-4B-Thinking.** All methods use a 20% visual reserve with 10% visual tokens exposed per decoding step.

Visual Selection	ChartQA		MMMU		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens						
Vanilla	499	87.4	4488	69.6	2494 0.0%	78.5 0.0%
Retain 10% Tokens						
Random	991	60.5	4097	58.6	2544 +2.0%	59.6 -24.1%
Fixed	936	61.2	3933	56.1	2435 -2.4%	58.7 -25.3%
w/o Trajectory	**673**	70.4	3892	58.0	2283 -8.5%	64.2 -18.2%
All-visible	695	**71.3**	3893	59.8	2294 -8.0%	65.6 -16.5%
VisionPulse ICML26	1043	70.0	4448	57.0	2746 +10.1%	63.5 -19.1%
VISTA (Ours)	698	**71.3**	**3844**	**60.9**	**2271** -8.9%	**66.1** -15.8%

Evidence Revisit Strategy During Decoding. To further examine attention-trajectory-guided evidence revisit, we compare different visual selection strategies under the same 20% visual reserve, as shown in Table 5. At each decoding step, only 10% of visual tokens are exposed. Fixed keeps the initial subset unchanged, Random resamples visible tokens at each step, and w/o Trajectory uses only current attention without historical information. We also include VisionPulse (Xu et al., 2026) as a single-step dynamic decoding baseline. VISTA improves average accuracy by 7.4 and 6.5 percentage points over Fixed and Random, respectively, while producing shorter generations. Removing the attention trajectory reduces accuracy from 66.1% to 64.2%, while VisionPulse achieves 63.5% with substantially longer outputs, showing the benefit of combining historical attention with immediate visual demand. VISTA also matches All-visible performance, achieving 66.1% versus 65.6% accuracy while exposing only a subset of the reserve at each step.

4 Conclusion

In this work, we revisit the practical efficiency of visual token pruning for reasoning-capable VLMs. We observed that reducing visual tokens does not necessarily translate into proportional end-to-end efficiency gains, as aggressive visual compression can reshape the subsequent reasoning process. Missing evidence and fragmented visual context may increase uncertainty, trigger repeated verification, and substantially inflate generation length, revealing a close coupling between visual compression and decoding that is overlooked by existing pruning methods. Motivated by this observation, we propose VISTA, a training-free framework that aligns visual token compression with the evolving evidence needs during decoding. Extensive experiments across architectures and diverse datasets demonstrate that explicitly accounting for this interaction improves both task performance and generation efficiency under strict visual budgets. Moreover, VISTA's evidence-revisit and reasoning-termination mechanisms remain effective when integrated with existing pruning methods, suggesting that these benefits are not tied to a particular token selection strategy. More broadly, our findings suggest that visual token reduction and downstream reasoning should not be treated as independent stages. As VLMs become stronger reasoners, coordinating visual compression with decoding-time computation may offer a promising direction toward more effective end-to-end inference.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript polishing, literature search, and code optimization and debugging. We did not use generative AI tools for the core research methodology, experimental design, or interpretation of results. We have reviewed all AI-assisted work. All AI-assisted code was manually reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work primarily focuses on improving the inference efficiency of vision-language models. Experiments are conducted on publicly available benchmarks and a private, de-identified medical VQA dataset collected under Institutional Review Board (IRB) approval and used in accordance with relevant institutional and data-use requirements.

REPRODUCIBILITY STATEMENT

We provide detailed implementation settings, hyperparameters, and experimental protocols in the main paper and appendix. Models and most datasets used in our experiments are publicly available. We also provide an anonymous code repository in Appendix A, which contains the implementation needed to reproduce our results.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

Sixun Dong, Juhua Hu, Mian Zhang, Ming Yin, Yanjie Fu, and Qi Qian. Mmtok: Multimodal coverage maximization for efficient inference of vlms. In *International Conference on Learning Representations*, volume 2026, pp. 48075–48099, 2026.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *arXiv preprint arXiv:2603.09480*, 2026.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Accelerating vision-language models through dynamic token sparsity. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20477–20486. IEEE, 2025.

Runhui Huang, Xinpeng Ding, Chunwei Wang, Jianhua Han, Yulong Liu, Hengshuang Zhao, Hang Xu, Lu Hou, Wei Zhang, and Xiaodan Liang. Hires-llava: Restoring fragmentation input in high-resolution large vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29814–29824. IEEE, 2025.

Jiwan Kim, Kibum Kim, Wonjoong Kim, Byung-Kwan Lee, and Chanyoung Park. Why and when visual token pruning fails? a study on relevant visual information shift in mllms decoding. In *European Conference on Computer Vision*, pp. 244–263. Springer, 2026a.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. Zoo-prune: training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

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

Matthew Renze. The effect of sampling temperature on problem solving in large language models. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 7346–7356, 2024.

Zhichao Sun, Yidong Ma, Gang Liu, Yibo Chen, Xu Tang, Yao Hu, and Yongchao Xu. Ivcprune: Revealing the implicit visual coordinates in lvlms for vision token pruning. *arXiv preprint arXiv:2602.03060*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

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

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

Hengbo Xu, Shengjie Jin, Yanbiao Ma, and Zhiwu Lu. Visionpulse: Dynamic visual sparsity for efficient multimodal reasoning. *arXiv preprint arXiv:2605.31457*, 2026.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24972–24982. IEEE, 2025.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free mllm acceleration. *arXiv preprint arXiv:2601.22674*, 2026.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

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

A Dataset and Implementation

A.1 Dataset Details

Dataset profile. We evaluate on seven multimodal VQA datasets, including six public benchmarks, ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), MathVista (Lu et al., 2024), MMMU (Yue et al., 2024), and ScienceQA (Lu et al., 2022), together with a private kidney pathology dataset, RenalVQA. These datasets cover a diverse range of tasks, including chart understanding, document and infographic reading, mathematical reasoning, multi-disciplinary knowledge, science question answering, and medical visual question answering, while also varying substantially in image complexity and visual token count.

For ChartQA (Masry et al., 2022), we use the full test split with 2,500 questions and a median of 425 visual tokens. We adopt the official relaxed accuracy, allowing a 5% relative error for numerical answers and exact matching otherwise. For DocVQA (Mathew et al., 2021), we use the full validation split with 5,349 questions, with a median of 3,710 visual tokens and a maximum of 16,280. For InfoVQA (Mathew et al., 2022), we use the full validation split with 2,801 questions over 483 images and an average of 3,482 visual tokens. Both DocVQA and InfoVQA are evaluated using ANLS, with similarities below 0.5 assigned zero score.

For MathVista (Lu et al., 2024), we use the full testmini split with 1,000 questions, including 540 multiple-choice and 460 free-form questions, with a median of 192 visual tokens. We follow the official answer extraction and normalization rules before exact matching. For MMMU (Yue et al., 2024), we use the validation split after removing 53 open-ended and 42 multi-image questions, resulting in 805 examples with a median of 204 visual tokens. For ScienceQA (Lu et al., 2022), we retain 2,017 image-based questions from the test split, with a median of 108 visual tokens. Both MMMU and ScienceQA are evaluated by exact matching of option letters.

RenalVQA is a private, de-identified medical VQA dataset built from renal biopsy archives collected between 2004 and 2006, containing 1,000 images from 630 patients with a median of 588 visual tokens. Questions and answers are generated from original medical records using generative AI and a standardized domain-specific terminology set, followed by pathologist review. The dataset was collected under Institutional Review Board (IRB) approval and used in accordance with relevant institutional and data-use requirements.

Code and Data Availability. All VLM weights and six public benchmarks used in our experiments are publicly available. We additionally provide an anonymous code repository at `https://anonymous.4open.science/r/VISTA-078B`, containing the implementation and scripts required to reproduce our experiments.

A.2 Implementation Details

Model Decoding. For Qwen3-VL-4B-Thinking and Qwen3-VL-8B-Thinking, we immediately close the thinking block opened by the official chat template before generation. Specifically, we append only a closing `</think>` tag followed by two newline characters to the end of the templated input. As a result, generation starts outside the thinking block rather than being forced to begin with a predefined explicit reasoning phase. Apart from this closing tag, we do not modify the official chat template or dataset-specific prompts. This setting does not prevent reasoning during generation, as the model may still re-enter a thinking block whenever needed. For Qwen3-VL-4B/8B-Thinking, we use a temperature of 1.0, top-p of 0.95, top-k of 20, and a maximum generation length of 16,384 tokens. For InternVL3.5-4B, we adopt the official Thinking system prompt and VLMEvalKit configuration, with a temperature of 0.6, top-p of 0.95, top-k of 50, a maximum generation length of 16,384 tokens, and a maximum context length of 24,576 tokens. Within each model family, all methods use the same prompts, sampling parameters, and generation limits.

Prompt and Answer Probe. Each example uses the original question and official dataset prompt, followed by the unified instruction `Please put your final answer within \boxed{}`. For the answer probe in VISTA, we append the temporary string `\n\n**Final Answer**\n\boxed{`. Under the Qwen3-VL tokenizer, this prompt corresponds to exactly 8

tokens. After appending the probe to the current decoding context, we read only the next-token distribution at its final position to estimate the model's current answer tendency. The KV-cache entries introduced by the probe are then removed immediately, so the probe does not persistently alter subsequent decoding.

Answer Extraction and Evaluation. We first extract the content of the last `\boxed{}` expression in the model output as the final answer. If no boxed answer is present, we apply a task-dependent fallback. For multiple-choice questions, we extract the option letter from the final answer sentence. For short-answer datasets, we locate the prediction according to the type of the ground-truth answer, such as taking the last numerical value for numeric questions, extracting yes/no for binary questions, and using the answer text for textual responses. The extracted prediction is then normalized and evaluated following the official protocol of each dataset.

Generation Length and Computational Cost. All experiments are conducted using the same Hugging Face inference engine on 8 NVIDIA A6000 GPUs. The average generation length L is defined as the number of generated tokens per example, including both intermediate reasoning and the final answer. Runtime efficiency is measured using per-example wall-clock time and LLM-side TFLOPs under the same inference engine. To analyze end-to-end inference cost, we further decompose computation into three components: the prefill cost associated with the number of visual tokens, the per-step decoding cost, and the accumulated decoding cost determined by the actual generation length. Detailed results for Qwen3-VL-4B-Thinking under 10% visual token retention are reported in Figure S1.

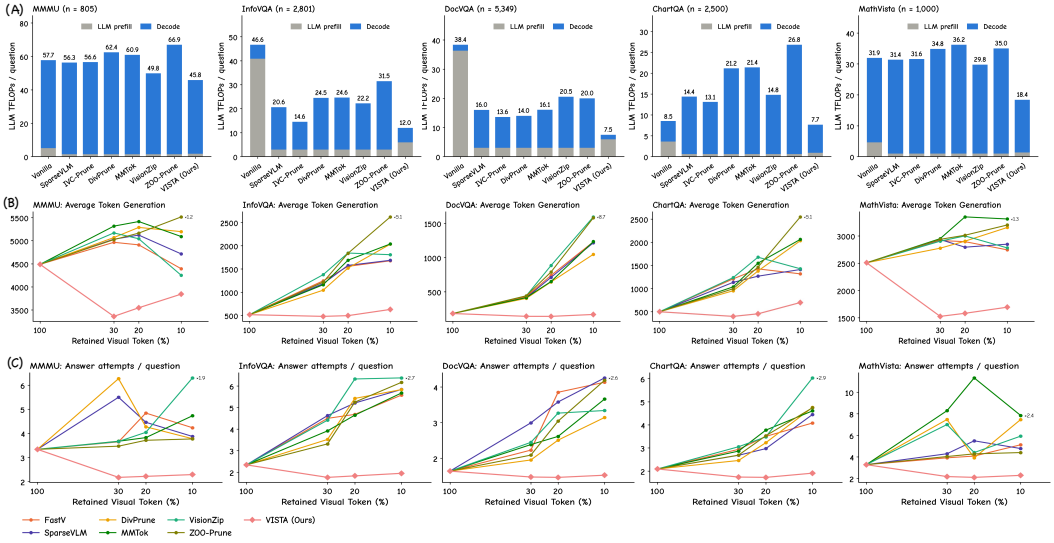

Figure S1: **Cost shift across datasets on Qwen3-VL-4B-Thinking at 10% visual token retention.** (A) reports prefill and decoding costs for each dataset. (B) and (C) show generation length and answer attempts across retention ratios, revealing increased reasoning overhead under more aggressive pruning.

B Additional Results

Generation length distribution and efficiency. Figure S2 compares Qwen3-VL-4B-Thinking under a 10% visual token budget across pruning baselines and VISTA in terms of generation length, answer attempts, and re-check frequency, together with the cumulative distribution of generated tokens. Compared with static pruning baselines, VISTA substantially reduces both the median generation length and repeated answer attempts, while bringing re-check behavior close to the unpruned model. The CDF further shows that VISTA shifts the overall generation-length distribution toward shorter responses, rather than benefiting only a small subset of examples. By combining dynamic evidence revisit with consistency-based answer probing, VISTA reduces unnecessary verification and prolonged reasoning, yielding more efficient decoding under the same visual token budget.

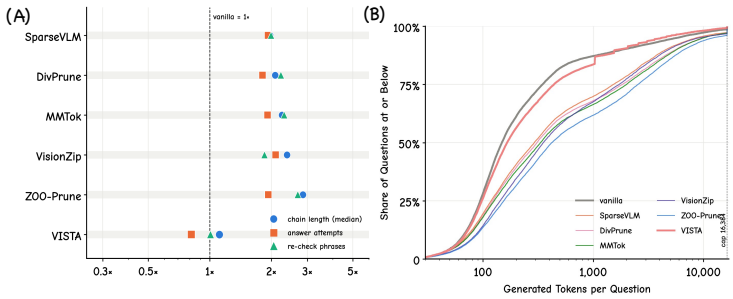

Figure S2: **Generation behavior and efficiency under a 10% visual token budget.** (A) Relative median generation length, answer attempts, and re-check frequency on Qwen3-VL-4B-Thinking, normalized by Vanilla. (B) Cumulative distribution of generated tokens per question. VISTA reduces repeated reasoning and shifts the generation-length distribution toward shorter outputs.

Table S1: **End-to-end runtime and LLM computational cost at a 10% visual token budget.** Runtime is measured in seconds per question (s/q) on Qwen3-VL-4B using the same Hugging Face runner. "Relative to VISTA" is the ratio of average runtimes. Avg LLM TFLOPs/q includes both prefill and decoding stage.

Method	Runtime (s/q) ↓					Avg (s/q) ↓	Relative to VISTA	Avg LLM TFLOPs/q ↓
	ChartQA	InfoVQA	MathVista	MMMU	RenalQA			
DivPrune CVPR25	19.4	28.4	34.7	58.1	8.5	29.8	1.63×	29.9
ZOO-Prune CVPR26	30.5	35.5	37.9	61.4	10.8	35.2	1.93×	33.8
VisionZip CVPR25	32.5	35.6	44.9	64.6	9.0	37.3	2.04×	24.8
MMTok ICLR26	30.3	23.6	37.7	59.8	8.1	31.9	1.75×	29.9
VISTA (Ours)	**12.1**	**9.0**	**21.7**	**42.4**	**6.2**	**18.3**	**1.00×**	**18.1**

Distinct behaviors under different pruning strategies. Different visual token pruning strategies can induce distinct generation behaviors, as illustrated in Figure S3. For diversity-based or text-guided methods such as DivPrune, the retained tokens are often scattered across the image, leaving fragmented visual evidence. This fragmented context can cause the model to repeatedly inspect the remaining evidence and make multiple answer attempts, substantially prolonging generation. In contrast, attention-based methods such as FastV tend to retain tokens concentrated in a few high-attention regions. When the complete answer-relevant region is not preserved, the model may lack sufficient evidence for further verification, form an incorrect judgment early, and terminate with a wrong answer. Although these methods exhibit different failure patterns, their common limitation is that a fixed visual subset cannot recover missing evidence as the reasoning process evolves.

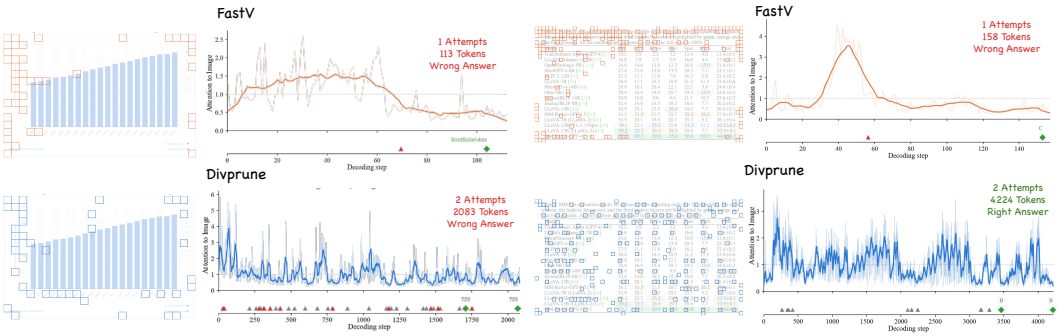

Figure S3: **Representative generation behaviors under different pruning strategies.** Two representative examples illustrate the retained visual tokens and attention trajectories of FastV and DivPrune. FastV may form an incorrect judgment early when the complete answer-relevant region is not preserved, while DivPrune may trigger repeated verification, multiple answer attempts, and substantially longer reasoning due to fragmented visual evidence.

Comprehensive performance comparisons. Due to space limitations in the main manuscript, we provide the full results on Qwen3-VL-8B-Thinking and InternVL3.5-4B in Tables S2 and S3, respectively, including both accuracy and average generation length across all datasets and visual token budgets. Across both model families and token budgets, VISTA consistently achieves a better balance between task performance and generation efficiency than existing pruning baselines.

Table S2: **Performance and generation efficiency on Qwen3-VL-8B-Thinking.** L denotes average generated length, and Acc denotes accuracy (%). Colored percentages indicate changes relative to Vanilla.

Method	ChartQA		DocVQA		InfoVQA		MathVista		MMMU		RenalQA		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retained 100% Tokens														
Vanilla	369	87.3	168	90.9	366	79.0	1477	77.1	3039	68.3	1686	46.5	1184 0.0%	74.9 0.0%
Retained 20% Tokens														
FastV ECCV24	1212	50.6	606	71.3	1163	37.7	2096	51.8	3823	57.9	2104	40.7	1834 +54.9%	51.7 -31.0%
SparseVLM ICML25	1090	53.8	376	80.1	1063	45.0	1910	53.9	3881	59.6	2122	40.9	1740 +47.0%	55.5 -25.8%
DivPrune CVPR25	924	65.1	406	80.8	1036	49.6	2081	59.0	3903	60.2	2033	41.2	1730 +46.1%	59.3 -20.8%
VisionZip CVPR25	881	68.4	332	84.1	911	55.3	2062	57.9	3698	62.9	1997	43.0	1647 +39.1%	61.9 -17.3%
MMTok ICLR26	975	61.2	412	79.9	1000	50.6	2068	52.2	3990	59.9	1968	41.0	1736 +46.6%	57.5 -23.2%
ZOO-Prune CVPR26	1017	64.8	545	76.2	1242	47.9	2004	58.8	3862	61.6	2014	42.5	1781 +50.4%	58.6 -21.7%
VISTA (Ours)	**351**	**79.8**	**145**	**90.4**	**378**	**71.3**	**1169**	**66.4**	**2550**	**64.2**	**1118**	**45.8**	**952** -19.6%	**69.6** -6.9%
Retained 10% Tokens														
FastV ECCV24	1284	29.0	1042	45.1	1197	25.2	1959	46.5	3462	54.7	2234	34.4	1863 +57.3%	39.2 -47.7%
SparseVLM ICML25	1143	32.8	735	60.3	1227	29.8	2038	45.7	3625	56.8	2214	37.6	1830 +54.6%	43.8 -41.4%
DivPrune CVPR25	1409	48.3	640	70.1	1362	38.2	2281	50.8	4083	58.9	2215	39.9	1998 +68.8%	51.0 -31.8%
VisionZip CVPR25	1443	43.5	746	64.8	1362	36.6	2066	46.0	3811	58.8	2105	39.9	1922 +62.3%	48.3 -35.5%
MMTok ICLR26	1199	37.8	783	67.4	1232	38.8	2322	43.8	3743	55.9	2085	39.2	1894 +59.9%	47.2 -37.0%
ZOO-Prune CVPR26	1312	42.0	1064	53.7	2031	29.8	2160	49.9	4250	56.6	2175	39.7	2165 +82.9%	45.3 -39.5%
VISTA (Ours)	**530**	**66.4**	**163**	**87.9**	**438**	**57.7**	**1305**	**57.5**	**2948**	**61.1**	**1145**	**43.4**	**1088** -8.1%	**62.3** -16.7%

Table S3: **Performance and generation efficiency on InternVL3.5-4B.** L denotes average generated length, and Acc denotes accuracy (%). Colored percentages indicate changes relative to Vanilla.

Method	ChartQA		DocVQA		InfoVQA		MathVista		MMMU		RenalQA		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retained 100% Tokens														
Vanilla	1656	88.2	980	86.9	3190	71.0	4511	74.8	6070	69.2	2802	50.9	3202 0.0%	73.5 0.0%
Retained 20% Tokens														
FastV ECCV24	2714	72.3	2735	60.0	5446	40.2	5142	60.0	6425	64.0	2726	47.7	4198 +31.1%	57.4 -22.0%
SparseVLM ICML25	2805	70.6	2436	62.9	5003	40.6	4879	62.1	6179	**68.5**	2710	**49.2**	4002 +25.0%	59.0 -19.8%
DivPrune CVPR25	3948	44.6	3657	30.2	5164	20.2	5407	55.3	6975	55.4	2297	45.3	4575 +42.9%	41.8 -43.1%
VisionZip CVPR25	4355	36.8	4068	36.0	5640	24.1	5312	51.4	6654	59.2	2591	43.8	4770 +49.0%	41.9 -43.0%
MMTok ICLR26	3477	44.8	3861	31.3	4675	22.5	5159	52.0	6693	61.2	2272	46.6	4356 +36.1%	43.1 -41.4%
ZOO-Prune CVPR26	4034	49.8	3482	32.2	4735	21.4	5304	54.8	6853	58.1	2669	45.9	4513 +41.0%	43.7 -40.5%
VISTA (Ours)	**615**	**79.8**	**697**	**75.4**	**1110**	**44.9**	**1456**	**66.2**	**2305**	65.0	**496**	48.1	**1113** -65.2%	**63.2** -14.0%
Retained 10% Tokens														
FastV ECCV24	3590	47.4	3654	35.7	5493	25.7	5049	51.4	6895	61.1	2419	46.4	4517 +41.1%	44.6 -39.3%
SparseVLM ICML25	3810	41.4	3448	39.0	5052	27.3	5130	**57.1**	6594	62.9	2443	**46.9**	4413 +37.8%	45.8 -37.7%
DivPrune CVPR25	3026	28.4	3945	17.7	3924	18.4	5000	47.7	6796	56.1	2705	44.1	4233 +32.2%	35.4 -51.8%
VisionZip CVPR25	3034	20.3	4847	13.0	3889	18.3	4905	41.7	6796	53.4	3181	40.7	4442 +38.7%	31.2 -57.5%
MMTok ICLR26	2723	27.7	3756	19.1	3759	20.0	5000	47.0	6760	55.3	2461	42.6	4077 +27.3%	35.3 -52.0%
ZOO-Prune CVPR26	3455	31.2	3346	20.1	4168	19.5	5249	47.8	6923	57.3	2934	42.2	4346 +35.7%	36.4 -50.5%
VISTA (Ours)	**928**	**55.4**	**854**	**59.9**	**996**	**41.8**	**1605**	55.5	**2483**	**64.2**	**501**	46.3	**1228** -61.6%	**53.9** -26.7%

Comprehensive ablation study. Due to space limitations in the main manuscript, we provide the full ablation results on Qwen3-VL-4B-Thinking in Tables S7 and S4, covering both the contributions of individual VISTA functional blocks and the generalization of its decoding mechanisms to existing pruning methods. The results consistently show that each component contributes to improved accuracy and generation efficiency, while integrating VISTA decoding with different pruning baselines substantially reduces generation length and recovers task performance across visual token budgets.

Effect of temperature setting. The preset sampling temperature of reasoning models can affect generation length (Bai et al., 2025; Renze, 2024). We therefore evaluate all methods under dif-

Table S4: **Effect of VISTA decoding across visual token budgets on Qwen3-VL-4B-Thinking.** L denotes average generated length, and Acc denotes accuracy (%). Percentages in parentheses indicate changes relative to Vanilla.

Method	ChartQA		InfoVQA		MathVista		MMMU		RenalQA		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens												
Vanilla	499	87.4	516	77.7	2508	75.6	4488	69.6	613	41.8	1725 0.0%	70.4 0.0%
Retain 20% Tokens												
DivPrune CVPR25	1379	67.2	1520	50.9	2903	58.4	5288	59.6	910	36.1	2400 +39.1%	54.4 -22.7%
DivPrune + VISTA	484	**81.8**	600	66.7	**1540**	66.7	3562	63.6	**512**	37.3	1340 -22.3%	63.2 -10.2%
VisionZip CVPR25	1677	47.1	1843	36.8	2999	46.9	5042	54.2	899	35.0	2492 +44.5%	44.0 -37.5%
VisionZip + VISTA	506	77.7	620	64.3	1726	63.2	3740	59.8	535	38.9	1425 -17.4%	60.8 -13.7%
MMTok ICLR26	1548	61.3	1689	49.1	3348	53.5	5415	57.3	889	34.6	2578 +49.5%	51.2 -27.4%
MMTok + VISTA	472	79.3	662	66.5	1611	65.0	3859	62.2	531	36.7	1427 -17.3%	61.9 -12.0%
Static VISTA	1087	71.0	1002	56.8	2902	60.3	5095	60.1	597	39.4	2137 +23.9%	57.5 -18.3%
VISTA (Ours)	**455**	79.0	**496**	**67.9**	1583	**68.0**	**3548**	**65.0**	579	**41.2**	**1332** -22.8%	**64.2** -8.8%
Retain 10% Tokens												
DivPrune CVPR25	2033	49.5	2034	35.1	3154	48.2	5196	53.7	1057	34.1	2695 +56.2%	44.1 -37.3%
DivPrune + VISTA	761	67.4	936	50.3	1757	58.5	3837	58.8	540	35.4	1566 -9.2%	54.1 -23.2%
VisionZip CVPR25	1427	20.4	1807	20.9	2780	41.7	4250	51.4	1056	32.8	2264 +31.3%	33.4 -52.5%
VisionZip + VISTA	1010	47.8	1173	38.0	1716	46.4	**3390**	56.8	571	34.8	1572 -8.9%	44.8 -36.4%
MMTok ICLR26	2062	38.6	2039	34.0	3310	42.9	5090	54.8	1018	31.3	2704 +56.8%	40.3 -42.7%
MMTok + VISTA	828	61.7	1037	48.6	1979	52.7	3788	57.6	**530**	35.3	1632 -5.4%	51.2 -27.3%
Static VISTA	1904	51.9	1438	45.0	3039	51.1	5292	56.5	582	39.2	2451 +42.1%	48.7 -30.8%
VISTA (Ours)	**698**	**71.3**	**629**	**56.7**	**1695**	**58.6**	3844	**60.9**	535	**39.6**	**1480** -14.2%	**57.4** -18.5%

ferent temperature settings. Lower temperatures generally lead to shorter typical reasoning chains. However, as shown in Table S5, reasoning inflation persists and can even become more pronounced under lower-temperature decoding. Across all methods and reported datasets, the median generation length decreases or remains nearly unchanged when reducing the temperature from 1.0 to 0.7, whereas the mean generation length increases. This discrepancy indicates that the increase is driven by a subset of abnormally long generations rather than a uniform shift of the entire distribution. Consistently, the proportion of responses falling into repetitive loops rises substantially at the lower temperature, especially for pruning baselines. For example, the average loop rate increases from 2.5% to 10.0% for SparseVLM and from 4.0% to 11.7% for DivPrune, compared with 1.3% to 4.9% for Vanilla. These results suggest that lower-temperature decoding makes it harder for the model to escape repeated re-examination of incomplete visual evidence after pruning, causing a small subset of samples to dominate the increase in average generation length.

Table S5: **Effect of decoding temperature on generation behavior on Qwen3-VL-4B-Thinking.** Mean and Median denote the mean and median generated length, respectively. Loop denotes the percentage of repetitive-loop responses. Average is computed over the three reported datasets.

Method	T	ChartQA				MathVista				MMMU				Average			
		Acc↑	Mean↓	Med.↓	Loop↓	Acc↑	Mean↓	Med.↓	Loop↓	Acc↑	Mean↓	Med.↓	Loop↓	Acc↑	Mean↓	Med.↓	Loop↓
Retained 100% Tokens																	
Vanilla	0.7	86.4	667	142	2.7	76.4	2529	384	3.9	68.0	4756	843	8.0	76.9	2651	456	4.9
	1.0	87.4	499	146	0.9	75.6	2508	464	0.9	69.6	4488	1069	2.2	77.5	2498	560	1.3
Retained 10% Tokens																	
SparseVLM	0.7	27.5	2446	235	12.6	45.0	3192	926	7.1	54.0	5189	975	10.4	42.2	3609	712	10.0
	1.0	28.6	1412	293	3.6	44.2	2848	984	2.1	54.2	4714	1311	1.9	42.3	2991	863	2.5
DivPrune	0.7	46.7	2886	456	16.0	49.9	3558	1018	8.2	56.1	5641	1463	10.8	50.9	4028	979	11.7
	1.0	49.5	2033	594	6.9	48.2	3154	1150	2.3	53.7	5196	1705	2.9	50.5	3461	1150	4.0
VISTA (Ours)	0.7	71.2	898	173	5.5	59.3	1817	504	5.6	58.6	4128	1034	8.9	63.0	2281	570	6.7
	1.0	71.3	698	180	2.2	58.6	1695	627	2.0	60.9	3844	1034	1.1	63.6	2079	614	1.8

Trade-off in visual reserve size. We examine the effect of different visual reserve sizes on Qwen3-VL-4B-Thinking while fixing the per-step visible budget to $c = 10\%$. As shown in Table S6, enlarging the reserve generally improves accuracy and reduces generation length by preserving more potentially useful visual evidence. However, a larger reserve also increases the prefill cost required

to cache additional visual tokens. We therefore use a reserve size of $2c$ as the default setting, which provides a favorable trade-off between retained visual evidence and end-to-end computational cost.

Table S6: **Effect of visual reserve size on Qwen3-VL-4B-Thinking with a 10% per-step visual budget.** L denotes average generation length, and TFLOPs measure total inference computation.

Reserve Size	InfoVQA			DocVQA			MMMU			Average		
	Acc↑	L↓	TFLOPs↓	Acc↑	L↓	TFLOPs↓	Acc↑	L↓	TFLOPs↓	Acc↑	L↓	TFLOPs↓
Reserve c	45.6	963	17.5	79.7	331	10.0	56.8	3900	46.7	60.7	1731	24.7
Reserve $1.5c$	53.3	708	16.2	86.1	198	10.0	57.9	4141	50.5	65.8	1682	25.6
Reserve $2c$	56.7	629	16.9	87.1	169	11.2	60.9	3844	46.4	68.2	1547	24.8
Reserve $3c$	59.6	534	23.4	88.8	147	17.0	62.0	3711	45.3	70.1	1464	28.6
Vanilla	77.7	516	96.1	90.4	183	72.9	69.6	4488	60.3	79.2	1729	76.4

Table S7: **Ablation of VISTA functional blocks on Qwen3-VL-4B-Thinking.** L denotes average generated length, and Acc denotes accuracy (%). A green square indicates that the corresponding block is enabled. Percentages in parentheses indicate changes relative to Vanilla.

Blocks			ChartQA		InfoVQA		DocVQA		MathVista		MMMU		ScienceQA		RenalQA		Average	
DS	ER	RT	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
Retain 100% Tokens																		
Vanilla			499	87.4	516	77.7	183	90.4	2508	75.6	4488	69.6	858	93.2	613	41.8	1381 0.0%	76.5 0.0%
Retain 10% Tokens																		
			2033	49.5	2034	35.1	1048	62.9	3154	48.2	5196	53.7	1087	82.7	594	33.7	2164 +56.7%	52.3 -31.7%
■			1904	51.9	1438	45.0	416	80.6	3039	51.1	5292	56.5	1125	82.9	592	37.2	1972 +42.8%	57.9 -24.4%
■	■		1088	71.2	962	56.1	222	87.2	2986	59.1	5223	60.6	1215	85.7	603	38.0	1757 +27.3%	65.4 -14.5%
	■	■	761	67.4	936	50.3	382	79.1	1757	58.5	3837	58.8	832	84.2	540	35.4	1292 -6.4%	62.0 -19.0%
■	■	■	**698**	**71.3**	**629**	**56.7**	**169**	**87.1**	**1695**	**58.6**	**3844**	**60.9**	**866**	**84.7**	**535**	**39.6**	**1205** -12.7%	**65.6** -14.3%

C Related Work

C.1 Large Vision-Language Models.

Large vision-language models (VLMs) have substantially advanced multimodal alignment and understanding. More recently, research on VLMs has shifted toward more complex multi-step reasoning. LLaVA-CoT (Xu et al., 2025), among others, explores explicit or structured multimodal reasoning processes. Meanwhile, modern VLMs increasingly adopt flexible visual representations to preserve fine-grained information. For instance, Qwen2-VL (Wang et al., 2024) introduces a dynamic-resolution mechanism that maps images of different sizes and aspect ratios into variable numbers of visual tokens, improving fine-grained visual modeling while substantially enlarging the visual context processed by the language model. Furthermore, post-training paradigms developed for reasoning LLMs have increasingly been extended to multimodal settings. Models such as Qwen3-VL (Bai et al., 2025) and InternVL3.5 (Wang et al., 2025) enhance complex visual reasoning through reinforcement-learning-based training, making modern VLM inference increasingly involve both long visual prefixes and substantial numbers of autoregressive reasoning tokens. This evolution also reshapes the efficiency bottleneck of multimodal reasoning. Although shortening the visual prefix can reduce prefill cost, the resulting information loss may induce additional search, verification, and self-correction during decoding, such that the end-to-end computational cost does not necessarily decrease accordingly.

C.2 Visual Token Pruning.

High-resolution images are typically encoded into hundreds of visual tokens, substantially increasing the prefill and KV cache costs. Recent training-free methods exploit redundancy in visual representations to retain only important tokens and reduce subsequent computation. For example, FastV (Chen et al., 2024), SparseVLM (Zhang et al., 2024), and PyramidDrop (Xing et al.,

2024) perform pruning based on attention responses, text relevance, and layer-wise visual redundancy, respectively, while VisionZip (Yang et al., 2025) further compresses visual representations by retaining dominant tokens and aggregating redundant contextual information. Another line of work places greater emphasis on the structure and coverage of the retained set. DivPrune (Alvar et al., 2025) promotes feature diversity, MMTok (Dong et al., 2026) selects tokens from the perspective of multimodal coverage, and ZOO-Prune (Kim et al., 2026b) combines output sensitivity with diversity-aware selection. Despite their different scoring and selection mechanisms, these methods primarily focus on determining which visual tokens should be retained, while the pruned visual set typically remains relatively static during subsequent generation. Once task-relevant evidence is removed at an early stage, additional text decoding cannot recover the discarded visual information.

For reasoning VLMs, such information loss may not only reduce accuracy but also induce repeated verification, answer revision, and longer reasoning processes, shifting part of the computation saved during prefill to the decoding stage. Therefore, visual token retention ratio or prefill cost alone cannot fully characterize end-to-end reasoning efficiency. A few recent studies (Xu et al., 2026; Kim et al., 2026a) have begun to dynamically adjust visual evidence during decoding. VisionPulse (Xu et al., 2026) update the visible token set according to evolving visual demands, suggesting that the visual evidence required during reasoning is not fixed. However, these methods mainly focus on dynamically adjusting visual tokens without jointly coordinating prefill cost, visual evidence access, and subsequent reasoning termination, making it difficult to reduce their combined computational overhead. In contrast, VISTA jointly considers visual compression, dynamic evidence recovery, and reasoning termination. Under a limited visual budget, it revisits relevant evidence on demand and reduces unnecessary continued generation once the answer stabilizes, extending visual token pruning from static input compression to end-to-end computation allocation for reasoning VLMs.

Table S8: **Performance across random seeds and visual token budgets.** L denotes average generated length, and Acc denotes accuracy (%). Average is computed over ChartQA, InfoVQA, MathVista, and MMMU. Colored percentages indicate changes relative to Vanilla.

Method	ChartQA		InfoVQA		MathVista		MMMU		Average	
	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑	L↓	Acc↑
100% Visual Tokens										
Vanilla	499	87.4	516	77.7	2508	75.6	4488	69.6	2003 0.0%	77.6 0.0%
20% Visual Tokens										
DivPrune CVPR25	1330±44	66.9±0.3	1498±46	51.0±0.2	3013±99	57.2±2.0	5229±89	59.8±0.3	2768 +38.2%	58.7 -24.3%
MMTok ICLR26	1520±71	61.1±0.6	1642±41	48.9±0.6	3241±97	53.4±0.1	5310±91	57.8±0.6	2928 +46.2%	55.3 -28.7%
VISTA (Ours)	**441±21**	**78.9±0.4**	**485±28**	**68.2±0.5**	**1584±33**	**67.5±0.4**	**3616±109**	**64.2±0.5**	**1532** -23.5%	**69.7** -10.2%
10% Visual Tokens										
DivPrune CVPR25	2082±47	48.9±0.7	2069±34	35.5±0.5	3113±62	48.7±0.8	5287±79	54.9±1.2	3138 +56.7%	47.0 -39.4%
MMTok ICLR26	2058±34	38.7±0.1	2009±47	34.3±0.3	3302±53	43.5±0.8	5003±101	54.5±0.7	3093 +54.4%	42.8 -44.9%
VISTA (Ours)	**711±17**	**71.1±0.3**	**646±21**	**56.9±0.3**	**1787±64**	**58.8±0.6**	**3875±48**	**59.8±0.7**	**1755** -12.4%	**61.7** -20.5%