ARE WE EVALUATING VLM TOKEN PRUNING RIGHT? LOOKING BEYOND ACCURACY AND TOKEN COUNT

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token pruning aims to reduce inference cost while preserving the capabilities of the original VLM. Yet existing evaluations rely on aggregate task accuracy and retained token count as proxies for pruning quality and actual efficiency. We reveal two limitations: 1) Aggregate accuracy can obscure whether a pruning method truly identifies critical visual tokens—many examples remain solvable under arbitrary token removal and models can exploit textual shortcuts when visual evidence is degraded. 2) Retained token count is an unreliable proxy for efficiency, as identical token budgets can incur different end-to-end costs depending on where and how pruning is performed. Consequently, comparisons based on aggregate accuracy and retained token counts fail to recover the global performance–efficiency Pareto frontier across methods. We introduce PAA-Bench, a ***plug-and-play*** evaluation framework for existing benchmarks, to evaluate whether pruning succeeds through meaningful token selection and whether this success translates into real speedup. To determine when token choice actually matters, we propose *Random-Pass@k* to measure how often an example remains correct under random pruning, enabling comprehensive evaluation across different levels of selection difficulty. To compare efficiency fairly, we introduce *Preservation-Constrained Speedup* (PCS) to report the maximum end-to-end speedup each method achieves under a shared preservation target. Experiments across *8* methods, *3* VLM backbones, and *9* benchmarks show that stronger token compression does not necessarily yield greater speedup, method differences widen on examples where token selection matters, and efficiency rankings change with the preservation target.

1 INTRODUCTION

Vision-language models (VLMs) are increasingly designed to process large and diverse visual inputs, including high-resolution images, multiple images, and videos (Lin et al., 2024; Wang et al., 2024). Supporting such inputs naturally requires longer visual-token sequences, increasing both computation and cache costs at inference (Su et al., 2026). This growth in token length makes efficient inference a key challenge for VLM deployment. However, enhancing efficiency through retraining requires substantial additional computation and data, and needs to be repeated when given new models. This makes training-free acceleration particularly attractive, as it can be applied directly to pretrained VLMs without model-specific retraining (Men et al., 2025; Zhang et al., 2025d).

Among training-free acceleration approaches, visual token pruning (Takezoe et al., 2026) directly shortens the visual sequence passed to later layers, avoiding computation on redundant visual information (Alvar et al., 2025; Zhang et al., 2025b). An effective pruning method should therefore translate this token reduction into actual inference acceleration while retaining the predictive capability of the original VLM. Accordingly, existing studies typically vary the number of visual tokens retained after pruning and report how model accuracy and inference time change as more tokens are removed (Yang et al., 2025a). Recent studies, such as G-Prune (Jiang et al., 2025) and FEATHER (Endo et al., 2025), have also illustrated that pruning sensitivity varies across tasks and benchmarks.

Despite these efforts, as illustrated in Figure 1, current evaluation can still provide an incomplete picture of pruning quality and realized efficiency. ***(i) Many benchmark examples do not truly test token-selection ability.*** We find that, across commonly used benchmarks such as GQA (Hudson & Manning, 2019), MMBench (Liu et al., 2024c), and POPE (Li et al., 2023), more than 80% of examples remain correct in 8–10 out of 10 *random-pruning* trials under most evaluated settings.

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.

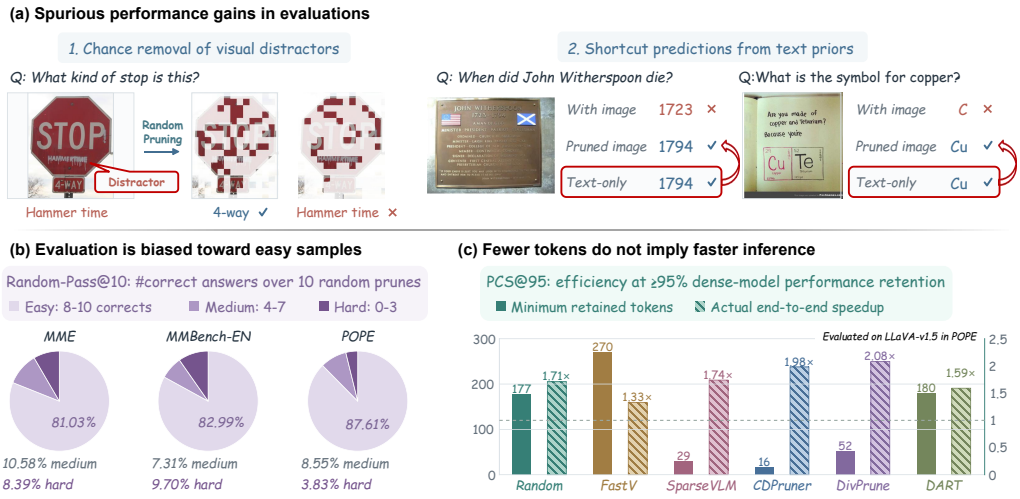


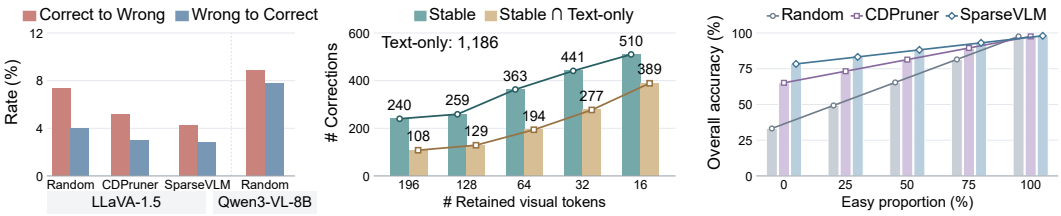
Figure 1: **Pitfalls in visual-token pruning evaluation.** (a) Corrections can arise from distractor removal or text priors and mask losses of originally correct predictions. (b) Under the budget of 64 tokens, easy examples dominate MME, MMBench-EN and POPE under Random-Pass@10. (c) At 95% correctness preservation, fewer retained tokens do not guarantee greater end-to-end speedup.

These pruning-insensitive examples can dominate aggregate results and obscure meaningful differences among token-selection methods. ***(ii) Aggregate accuracy does not directly measure capability preservation.*** Final accuracy mixes pruning-induced failures with wrong-to-correct transitions, which can arise from distractor removal or increased reliance on textual priors rather than better preservation of visual information. Such spurious performance gains can offset newly introduced errors and mask the loss of capabilities in the original VLM. ***(iii) Retained token count is an unreliable proxy for realized efficiency.*** Methods retaining the same number of tokens can incur substantially different end-to-end costs because pruning location and algorithmic overhead differ across methods. Together, these limitations show that token pruning evaluation should answer three distinct questions: "*whether performance depends on meaningful token selection, whether the model's capability is preserved, and how much realized speedup is achieved under that preservation level*".

To answer these questions, we introduce **PAA-Bench**, a *plug-and-play evaluation framework* for existing benchmarks that separates token-selection effectiveness, correctness preservation, and realized efficiency. Our central idea is to replace proxy-based comparison with direct measurements of whether token choice matters, what capability is preserved, and how much real speedup is obtained at that preservation level. Accordingly, we propose ***Random-Pass@k***, which uses repeated random pruning as a method-independent reference to determine whether successful predictions genuinely depend on token choice. To assess what pruning actually preserves, we then define performance rigorously by preservation of the dense model's correct predictions, so that spurious correction gains are not mistaken for preserved capability. Using capability preservation as the performance criterion, we introduce ***Preservation-Constrained Speedup*** (PCS), which reports each method's maximum measured end-to-end speedup under the same preservation target, including pruning overhead.

PAA-Bench therefore **shifts the basis of comparison from equal token counts to equal capability preservation, while separately revealing whether success actually depends on token selection**:

Main contributions. *(1)* We rethink conventional token-pruning evaluation and reveal that comparisons based on aggregate accuracy and retained token count can obscure whether success stems from meaningful token selection, whether the original model's capability is preserved, and whether compression translates into realized efficiency. *(2)* We introduce **PAA-Bench**, a plug-and-play framework for existing benchmarks that evaluates token-selection effectiveness, correctness preservation, and realized efficiency through Random-Pass@k, dense-model correctness preservation, and Preservation-Constrained Speedup, with Random-Pass@k theoretically characterizing the headroom available to informed selection from random-pruning success. *(3)* Across 8 pruning methods, 3 VLM backbones, and 9 benchmarks, PAA-Bench reveals a more discriminative view of the performance–efficiency trade-offs: preservation-aware scoring can reverse rankings, selection-sensitive examples expose larger method gaps, and efficiency rankings vary by preservation level.



(a) Correctness transitions (b) Correction counts (c) Dilution of method differences

Figure 2: **Diagnostics of token-pruning evaluation.** (a) **Correction gains can mask preservation losses**: correct-to-wrong and wrong-to-correct rates for LLaVA-1.5 at 128 tokens (left) and Qwen3-VL-8B at 20% token retention (right). (b) **Stable corrections overlap with text-only behavior**: counts of Stable, Text-only corrections, and their intersection across budgets on LLaVA-1.5. (c) **Method gaps widen as the Easy-example proportion decreases**: overall accuracy of Random, CDPruner, and SparseVLM across mixtures with 0–100% Easy examples. Details in Appendix G.1.

2 Related Work

Visual token reduction. Existing visual-token pruning methods (Li et al., 2025b; Vasu et al., 2025; Xing et al., 2025; Huang et al., 2025; Zhang et al., 2025c) use different signals to prune redundant tokens at different VLM inference stages (Aneja et al., 2026; Beyer et al., 2024; Team et al., 2025). FastV prunes visual tokens using attention scores from early language-model layers, whereas VisionZip preserves informative, complementary content before language-model processing (Chen et al., 2024; Yang et al., 2025a). DivPrune, DART, and SparseVLM instead emphasize diversity-, duplication-, or sparsification-based selection (Alvar et al., 2025; Wen et al., 2025; Zhang et al., 2025d). These methods are compared on post-pruning accuracy and inference speed at several fixed retained-token budgets (Hu et al., 2026; Deng et al., 2025). Yet a fixed budget does not define a comparable operating point: selectors may preserve different amounts of task-relevant information, while runtime depends on pruning stage, selection overhead, and computation incurred before pruning (Li et al., 2025a; Wang et al., 2025a; Chen et al., 2026; Sun et al., 2026; Yang et al., 2025b). Thus, equal token counts imply neither equal predictive performance nor equal computational cost, so fixed-budget comparisons may obscure the performance–efficiency trade-off across methods.

Evaluation of visual-token pruning. Recent work has attempted to improve this evaluation paradigm (Dong et al., 2026; Baek et al., 2026). VTC-Bench studies compression sensitivity by focusing on examples solvable by the dense model, while EffiVLM-Bench incorporates behavioral agreement and measured inference speed (Liao et al., 2026; Wang et al., 2025b). Nevertheless, aggregate scores can still mix correct-to-wrong losses with wrong-to-correct gains, and pruning-insensitive examples may dominate benchmark results. This leaves a gap for an evaluation protocol that directly connects token-selection quality, correctness preservation, and realized acceleration.

3 A Closer Look at Token-Pruning Benchmarks: Accuracy and Efficiency

We revisit visual token-pruning evaluation from its fundamental objective: identifying the method–configuration that maximizes realized inference speedup under a common performance requirement,

$$\max_{M\in\mathcal{M},\, c\in\mathcal{C}_M} \underbrace{\frac{T_{\text{orig}}}{T_p(M,c)}}_{\text{Speedup}(M,c)} \qquad \text{s.t.} \qquad \text{Acc}_{\mathcal{D}}(M,c) \geq \tau, \tag{1}$$

where $M \in \mathcal{M}$ indexes a pruning method and $c \in \mathcal{C}_M$ one of its evaluated configurations. T_{orig} and $T_p(M,c)$ are the measured runtimes of the original and pruned pipelines, with T_p including pruning overhead; $\text{Acc}_{\mathcal{D}}(M,c)$ is the post-pruning accuracy on $\mathcal{D}$, and τ the shared accuracy requirement.

This objective requires evaluating two complementary dimensions: ***accuracy*** *and* ***realized efficiency***. The validity of this comparison therefore depends on (1) whether $\text{Acc}_{\mathcal{D}}$ faithfully measures

capability preservation, on examples informative for token-selection quality, and (2) whether T_p reflects the realized cost of the complete pruning pipeline. We examine these requirements in turn.

3.1 On Accuracy: Preservation and Selection Discriminability

We first examine the accuracy side of Eq. 1. Existing token-pruning evaluations typically use aggregate post-pruning accuracy as the performance criterion. To understand what this score actually reflects, we decompose it according to how pruning changes the original model's predictions:

$$\mathrm{Acc}_{\mathcal{D}}(M,c) = \underbrace{\alpha_{\mathcal{D}}^{\mathrm{orig}}}_{\text{original accuracy}} - \underbrace{\alpha_{\mathcal{D}}^{\mathrm{orig}} r_{C\to W}^{M,c}}_{\text{correct}\to\text{wrong}} + \underbrace{\left(1-\alpha_{\mathcal{D}}^{\mathrm{orig}}\right) r_{W\to C}^{M,c}}_{\text{wrong}\to\text{correct}}. \tag{2}$$

Here, $\alpha_{\mathcal{D}}^{\mathrm{orig}}$ is the original-model accuracy, while $r_{C\to W}^{M,c}$ and $r_{W\to C}^{M,c}$ are the rates of originally correct predictions becoming wrong and originally wrong predictions becoming correct after pruning.

This decomposition explicitly illustrates how aggregate accuracy combines preservation losses and correction gains. Under this view, aggregate accuracy can fail in two distinct ways: corrections may inflate the score without reflecting capability preservation, while preserved predictions may still provide little evidence of effective token selection, as depicted below.

Challenge I: Wrong-to-correct gains can mask capability loss in aggregate accuracy. To quantify the transition terms in Eq. 2, we measure correct-to-wrong and wrong-to-correct rates on LLaVA-1.5 and Qwen3-VL-8B at 20% visual-token retention. As in Figure 2(a), wrong-to-correct transitions are of a similar magnitude to correct-to-wrong transitions across methods. On LLaVA-1.5, the wrong-to-correct contribution is about half to two-thirds as large as the correct-to-wrong contribution, while the two are even closer on Qwen3-VL-8B (7.76% vs. 8.85%). These results show that *correction gains can mask over half—and sometimes nearly all—of the accuracy loss*.

These correction gains, however, need not reflect better token selection. We identify two mechanisms through which such gains can arise:

1. *Accidental removal of visual distractors.* Pruning may incidentally remove visual evidence that mislead the original model. The stop-sign example in Figure 1(a) illustrates this case: pruning removes tokens that drive the original model toward an incorrect prediction, causing the pruned model to instead output the correct "4-way" answer. The correction therefore reflects incidental removal of misleading evidence rather than more effective visual evidence preservation.
2. *Increased reliance on textual priors.* As visual evidence is weakened, the model may rely more heavily on image-agnostic textual priors. We define stable corrections as examples that remain corrected across all random-pruning trials and compare them with a text-only control. As shown in Figure 2(b), both the number of stable corrections and their overlap with text-only-correct examples increase as fewer visual tokens are retained. In particular, the overlap rises from about 45% at 196 tokens to 76% at 16 tokens. These corrections therefore reflect increased reliance on image-agnostic textual priors, rather than visually grounded evidence.

Together, these mechanisms show that wrong-to-correct gains can arise without improved token selection. However, aggregate accuracy allows spurious correction gains to offset losses on originally correct predictions, thereby underestimating the capability lost after pruning. Capability preservation should therefore be evaluated independently of pruning-induced correction gains.

Challenge II: Pruning-insensitive examples obscure token-selection quality. Even after separating capability preservation from correction gains, some examples remain correct regardless of which visual tokens are retained, making their preserved correctness weak evidence of effective token selection. We therefore ask a counterfactual: would the model still answer correctly if the retained tokens were chosen at random? We study this sensitivity via Random-Pass@k (Eq. 5), which counts how many of k random-pruning trials preserve correctness for each originally correct example.

Proposition 1 (*High random-pruning success limits gains from informed selection*) *Consider M visual patches, N of which independently suffice for a correct prediction. If B patches are retained*

uniformly at random and correctness requires retaining at least one such patch, then

$$p_N = 1 - \frac{\binom{M-N}{B}}{\binom{M}{B}}, \qquad p_{N+1} \geq p_N, \qquad q_N - p_N \leq 1 - p_N, \tag{3}$$

where p_N and q_N denote the success probabilities of random and informed selection. Random pruning fails only when all B retained patches come from the $M-N$ non-evidence patches, yielding p_N. Increasing N reduces this failure probability, while $q_N \leq 1$ directly gives $q_N - p_N \leq 1 - p_N$.

Random-Pass@k is therefore grounded by Proposition 1, which shows that higher random-pruning success leaves less room for informed selection to improve. With $k = 10$, we accordingly group examples into Easy (8–10 random successes), Medium (4–7), and Hard (0–3).

As shown in Figure 1(b), we reveal that Easy examples constitute over 80% of MMBench-EN, POPE, and MME, causing pruning-insensitive samples to dilute differences among token selectors. Conversely, TextVQA contains 48.49% Hard examples (Appendix G.1), providing an explanation for its stronger discriminability observed in prior work. However, TextVQA covers only a narrow range of text-centric VLM capabilities, whereas ***Random-Pass@k enables broader benchmarks to be restructured into selection-sensitive evaluations*** while retaining the original capability coverage.

3.2 On Realized Efficiency

Challenge III: Fixed-token comparisons misrepresent realized efficiency across methods. Existing evaluations commonly compare methods at matched retained-token budgets. However, the same token budget does not place methods at comparable operating points. This creates two mismatches: *(i) accuracy*, because the same token budget can correspond to different preservation levels across methods; and *(ii) realized efficiency*, because identical token counts can incur different end-to-end costs depending on pruning location and selection overhead. Fixed-token comparisons can therefore misrepresent both predictive preservation and realized efficiency across methods.

Figure 1(c) confirms both mismatches. For preservation, CDPruner and FastV require only 16 and 270 visual tokens, respectively, to reach the same 95% correctness-preservation target, showing that the same preservation can require very different token budgets. For realized efficiency, CDPruner retains fewer tokens than DivPrune (16 *vs.* 52) but achieves a lower speedup ($1.98\times$ *vs.* $2.08\times$), revealing that token compression and realized speedup can rank methods in opposite orders.

Cross-method comparison should therefore match preservation rather than token count and measure realized end-to-end speedup directly. This leads to comparing the maximum realized speedup under a shared preservation target, introduced as *Preservation-Constrained Speedup* (PCS) in Section 4.2.

4 PAA-Bench: Evaluation Framework

The preceding analysis motivates an evaluation framework that separates token-selection quality from realized efficiency. Based on these observations, we introduce PAA-Bench with two complementary evaluation components: **Correctness Preservation and Selection-Difficulty Stratification** (Section 4.1) and **Preservation-Constrained Speedup** (Section 4.2).

4.1 Correctness Preservation and Selection-Difficulty Stratification

Section 3.1 shows that post-pruning correctness alone is insufficient to characterize token-selection quality: it may not represent preserved capability, and even preserved correctness may not be attributable to effective token selection, when the example itself is pruning-insensitive. We therefore evaluate token selection in two stages: first measuring correctness preservation on the original-correct subset, and then stratifying these examples by random-pruning sensitivity.

Correctness preservation. As shown in Section 3, aggregate post-pruning accuracy inherently conflates correct-to-wrong regressions with wrong-to-correct gains. Since the latter may reflect distractor-removal effects or textual-prior reliance, they can partially mask substantial losses of originally correct predictions. This motivates explicitly conditioning preservation on dense-model correctness, thereby excluding correction gains from the resulting metric by construction.

Algorithm 1 PAA-Bench evaluation protocol

Input: Evaluation set $\mathcal{D}$, original VLM f_D, pruning methods $\mathcal{M}$, random trials k, preservation target τ
Output: Difficulty subsets, correctness preservation, and $\{\mathrm{PCS}_M@\tau\}_{M\in\mathcal{M}}$
Stage 1: Construct the preservation set
$\mathcal{D}^+ \leftarrow \{x_i \in \mathcal{D} \mid f_D(x_i) = y_i\}$
Stage 2: Stratify token-selection difficulty
for each $x_i \in \mathcal{D}^+$ **do**
  Compute Random-Pass@$\,k(x_i)$ using k random-pruning trials
  Assign x_i to $\mathcal{D}_{\mathrm{Easy}}$, $\mathcal{D}_{\mathrm{Medium}}$, or $\mathcal{D}_{\mathrm{Hard}}$
end for
Stage 3: Compare realized efficiency at matched preservation
for each pruning method $M \in \mathcal{M}$ **do**
  Evaluate correctness preservation and realized speedup for each $c \in \mathcal{C}_M$
  $\mathrm{PCS}_M@\tau \leftarrow \max_{c\in\mathcal{C}_M} \mathrm{Speedup}(M,c) \quad \text{s.t.} \quad \mathrm{Acc}_{\mathcal{D}^+}(M,c) \geq \tau$
end for

Accordingly, let $\mathcal{D}^+ = \{(x_i, y_i) \in \mathcal{D} \mid f_D(x_i) = y_i\}$ denote this dense-correct subset. For pruning method M under configuration c, we define correctness preservation as

$$\mathrm{Acc}_{\mathcal{D}^+}(M,c) = \frac{\sum_{(x_i,y_i)\in\mathcal{D}^+} \mathbf{1}[f_{M,c}(x_i) = y_i]}{|\mathcal{D}^+|}. \tag{4}$$

This quantity measures the fraction of dense-correct examples that remain correct after pruning. Higher values therefore indicate that the pruned model preserves a larger share of the dense model's originally correct predictions, while wrong-to-correct transitions are excluded by construction.

Selection-difficulty stratification. After restricting evaluation to dense-correct examples, preservation alone still does not establish effective token selection, since some examples remain correct regardless of which visual tokens are retained. We therefore use repeated random pruning to calibrate per-example selection sensitivity. For each $(x_i, y_i) \in \mathcal{D}^+$, we conduct k independent trials under matched pruning settings and define

$$\text{Random-Pass@}\,k(x_i) = \sum_{j=1}^{k} \mathbf{1}\left[f_{m_j}(x_i) = y_i\right], \qquad (x_i, y_i) \in \mathcal{D}^+, \tag{5}$$

where m_j denotes the random pruning mask in trial j. Random-Pass@k ranges from 0 to k; lower scores indicate that random pruning is less likely to preserve correctness, creating more room for performance differences among informed selectors to emerge, whereas higher scores imply limited headroom and weaker discriminative value. With $k = 10$, we stratify examples into **Easy (8–10)**, **Medium (4–7)**, and **Hard (0–3)**, and report correctness preservation for each stratum. Together, correctness preservation quantifies retention of the dense model's originally correct predictions, while Random-Pass@k identifies selection-sensitive examples that better differentiate token selectors.

4.2 Preservation-Constrained Speedup

Having established what pruning preserves and when token choice matters, we next ask whether token reduction translates into realized efficiency. As shown in Section 3.2, retained-token count alone cannot characterize this trade-off, since methods differ in correctness preservation and end-to-end cost. We therefore compare complete pruning pipelines under a shared preservation target.

To this end, we introduce *Preservation-Constrained Speedup* (PCS), defined as the maximum measured end-to-end speedup over a method's evaluated configurations satisfying a shared preservation target. For method M, we define

$$\mathrm{PCS}_M@\tau = \max_{c\in\mathcal{C}_M} \frac{T_D}{T_{M,c}} \qquad \text{s.t.} \qquad \mathrm{Acc}_{\mathcal{D}^+}(M,c) \geq \tau, \tag{6}$$

where $\mathcal{C}_M$ denotes the set of evaluated configurations, and T_D and $T_{M,c}$ denote the measured end-to-end runtimes of the dense and pruned pipelines, respectively. Both are measured under matched hardware and inference settings, with $T_{M,c}$ covering the complete pruning pipeline, including token-selection and gathering overhead (Wang et al., 2025b). PCS@95 sets $\tau = 0.95$ and identifies the

fastest measured configuration that preserves at least 95% of the dense model's originally correct predictions, shifting comparison from common token budgets to common preservation levels.

5 EXPERIMENTS

Building on the preceding analysis of visual-token pruning evaluation, we conduct extensive experiments to systematically assess pruning methods across different evaluation criteria and quantify the efficiency gains realized in practical settings. We address the following research questions (***RQs***):

RQ1. *How do wrong-to-correct gains influence the relative rankings of token-pruning methods?*

RQ2. *Do performance differences among methods widen as token-selection difficulty increases?*

RQ3. *Which methods achieve the greatest realized speedup under a shared preservation target?*

Setup. Across three VLM backbones, LLaVA-1.5 (Liu et al., 2024a), LLaVA-NeXT (Liu et al., 2024b), and Qwen3-VL (Bai et al., 2025), we evaluate eight pruning methods: CDPruner (Zhang et al., 2025b), DivPrune (Alvar et al., 2025), FastV (Chen et al., 2024), VisionZip (Yang et al., 2025a), DART (Wen et al., 2025), PruMerge (Shang et al., 2025), SparseVLM (Zhang et al., 2025d), and VisPruner (Zhang et al., 2025a), with random pruning (Random) as a reference. Evaluation covers GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018), ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al., 2023), MME (Fu et al., 2025), MMBench (Liu et al., 2024c), and VQAv2 (Goyal et al., 2017), with method and task coverage varying by backbone.

Table 1: **Correctness preservation on LLaVA-1.5 (%).** Dataset scores average the three difficulty strata. S-Avg and W-Avg average stratum-weighted and sample-weighted scores across nine datasets. Rel. averages pruned-to-dense score ratios over the same nine datasets. Bold/underline: best/second-best per budget. Full results appear in Appendix D.

Method	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	VQA^{v2}	MMB	MMB^{CN}	S-Avg	W-Avg	Rel.
Retain 128 visual tokens												
Random	56.7	56.3	55.2	55.1	57.8	56.5	57.3	55.3	54.3	56.1	85.6	90.2
CDPruner	73.8	74.4	71.7	86.7	68.9	73.7	80.0	69.2	70.1	74.3	92.1	**98.3**
DivPrune	68.8	72.7	70.1	83.3	68.7	69.5	74.9	66.8	66.2	71.2	91.2	97.4
FastV	63.3	81.1	**85.2**	89.7	73.1	**83.2**	75.4	77.2	82.1	78.9	90.9	93.5
DART	72.8	83.8	85.0	86.1	79.7	81.3	80.9	**79.6**	83.3	81.4	93.0	96.6
SparseVLM	**74.4**	**84.1**	81.6	**92.3**	**79.8**	78.9	**83.6**	78.2	**85.6**	**82.1**	**94.4**	96.4
Retain 64 visual tokens												
Random	56.2	55.6	54.6	53.5	58.2	55.5	56.8	55.5	54.9	55.6	79.9	85.5
CDPruner	73.9	74.2	72.1	84.3	79.9	75.3	82.5	75.1	72.0	76.6	89.3	**96.9**
DivPrune	69.2	72.0	70.7	82.4	79.1	69.4	76.5	66.0	68.4	72.6	88.0	95.2
FastV	55.1	72.3	81.9	82.2	58.7	71.5	65.6	72.2	74.6	70.5	83.1	76.5
DART	66.6	77.4	80.3	79.6	71.5	80.2	75.5	75.4	78.1	76.1	87.8	93.5
SparseVLM	**76.5**	**84.1**	**86.1**	**92.0**	**82.7**	**83.0**	**86.8**	**86.1**	**90.1**	**85.3**	**93.3**	88.1
Retain 32 visual tokens												
Random	55.9	54.4	54.3	52.9	57.2	55.2	56.1	54.6	54.3	55.0	73.8	78.4
CDPruner	**74.4**	**73.5**	74.2	**80.7**	**82.9**	**74.0**	**82.9**	**75.0**	67.4	**76.1**	**85.5**	**94.9**
DivPrune	67.5	70.5	71.8	78.2	78.0	70.6	75.6	69.8	71.5	72.6	83.9	92.1
DART	61.7	69.0	**77.4**	71.7	68.8	73.0	68.6	70.7	**72.1**	70.3	80.7	89.6

5.1 METHOD RANKINGS UNDER DIFFERENT EVALUATION METRICS (RQ1)

To test whether wrong-to-correct gains alter method rankings, Table 1 compares RelAcc with W-Avg. RelAcc is the pruned-to-dense score ratio under each benchmark's original metric, whereas W-Avg measures correctness preservation on dense-correct examples, weighting difficulty strata by their sample counts within each dataset and averaging datasets equally. On LLaVA-1.5 with 64 retained visual tokens, CDPruner achieves a RelAcc of 96.9 versus SparseVLM's 88.1, but their W-Avg scores are 89.3 and 93.3, respectively, reversing the ranking. Thus, higher relative accuracy does not necessarily imply better preservation of the dense model's correct predictions.

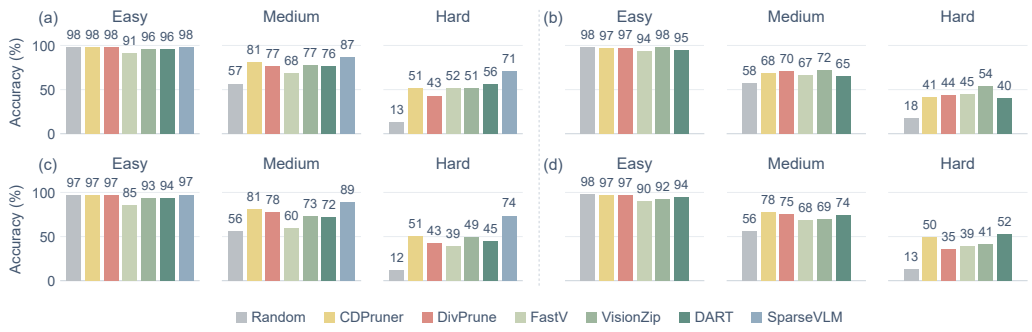


Figure 3: **Method differences widen with selection difficulty.** (a) LLaVA-1.5, 64 tokens; (b) Qwen3-VL, 30% retention; (c) LLaVA-1.5, 32 tokens; (d) LLaVA-NeXT, 160 tokens. Random reports the mean of 10 trials. Implementation details are in Appendix C.3.

The disagreement extends beyond the preceding example. Among the five methods reported under both metrics, five of ten pairwise orderings at 128 tokens and three at 64 tokens reverse between RelAcc and W-Avg. At both budgets, SparseVLM moves from fourth under RelAcc to first under W-Avg. These reversals highlight the diagnostic value of PAA-Bench: its preservation-aware evaluation reveals substantial ranking differences obscured by conventional aggregate task scores.

These ranking shifts reflect a fundamental distinction between the two evaluation metrics. RelAcc allows wrong-to-correct gains to offset correct-to-wrong losses in the benchmark score, whereas W-Avg conditions on dense-correct examples and directly quantifies correctness preservation. RelAcc therefore summarizes net task performance, while W-Avg explicitly reveals pruning-induced losses of originally correct predictions; the former should not be treated as a reliable proxy for the latter.

5.2 Discriminability across Selection Difficulty (RQ2)

To assess how selection difficulty affects method separation, Figure 3 compares correctness preservation across Easy, Medium, and Hard examples. Across all four evaluated settings, the gap in correctness preservation among methods widens with difficulty. We define separation as the difference between the highest and lowest scores among the displayed pruning methods, excluding Random. On LLaVA-1.5 with 64 retained visual tokens, this gap increases from 6.9 percentage points on Easy examples to 18.8 on Medium and 27.6 on Hard examples. Taken together, these results show that harder examples provide a more discriminative evaluation of token-selection strategies.

Tables 1 and 2 extend the difficulty-aware comparison across vision-language tasks using W-Avg and S-Avg. Both weight datasets equally, but within each dataset W-Avg preserves empirical stratum frequencies, whereas S-Avg weights Easy, Medium, and Hard equally. For LLaVA-NeXT at 320 retained visual tokens, the gap between Random and DART increases from 7.5 percentage points under W-Avg (84.8 vs. 92.3) to 25.5 under S-Avg (56.0 vs. 81.5); the gap between VisionZip and Random similarly grows from 4.2 to 14.4 points. When Easy examples dominate the sample mix, empirical weighting can therefore obscure method separation in the selection-sensitive strata. By reporting both aggregates, PAA-Bench makes these difficulty-dependent differences more visible.

5.3 Realized Speedup under Preservation Constraints (RQ3)

We use PCS to compare measured end-to-end speedup under shared correctness-preservation targets. Figure 4 compares measured results across methods, with complete results provided in Appendix E. The leading method varies across models and tasks. At 95% preservation, CDPruner achieves the highest speedup of $4.37\times$ on LLaVA-NeXT/POPE, whereas SparseVLM leads on Qwen3-VL/TextVQA with $1.23\times$ and reaches $1.31\times$ at 90% preservation.

Greater token compression does not necessarily yield greater realized speedup. On LLaVA-1.5/MMBench at 90% preservation, SparseVLM retains only 4 visual tokens and achieves a $1.83\times$ speedup, whereas DivPrune retains 49 visual tokens yet achieves a higher speedup of $1.95\times$. Pruning may even fail to produce a net acceleration: on Qwen3-VL/TextVQA at 95% preservation, VisionZip retains 289 visual tokens but achieves a speedup of only $0.95\times$. As discussed in Section 3.2, run-

Table 2: **Correctness preservation on LLaVA-NeXT (%).** Selected methods at 640 and 320 retained visual tokens. Dataset scores average the three difficulty strata; S-Avg and W-Avg average stratum-weighted and sample-weighted scores across eight datasets. Bold/underline: best/second-best among the shown methods per budget. Full results appear in Appendix D.

Method	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	MMB	MMB^{CN}	S-Avg	W-Avg	Rel.
Retain 640 visual tokens											
Random	57.2	56.0	56.9	55.3	59.1	58.0	56.3	56.3	56.9	89.6	93.6
DART	**83.2**	**86.8**	**86.0**	**90.3**	**89.1**	**86.3**	**80.7**	**87.6**	**86.3**	**96.1**	99.1
DivPrune	62.5	69.8	68.7	75.1	75.0	75.1	67.9	58.9	69.1	91.7	99.0
VisionZip	64.7	75.1	63.8	86.0	73.5	72.5	73.2	66.3	71.9	92.7	**100.1**
Retain 320 visual tokens											
Random	56.6	55.0	56.2	53.7	58.3	56.2	57.1	55.2	56.0	84.8	89.7
DART	**77.4**	**82.1**	**77.6**	**85.4**	**81.7**	**82.6**	**79.1**	**85.9**	**81.5**	**92.3**	97.0
DivPrune	63.8	69.4	62.1	74.5	80.4	70.6	73.7	64.6	69.9	88.9	**97.3**
VisionZip	63.5	75.6	61.2	85.0	75.1	68.4	70.4	63.9	70.4	89.0	96.8

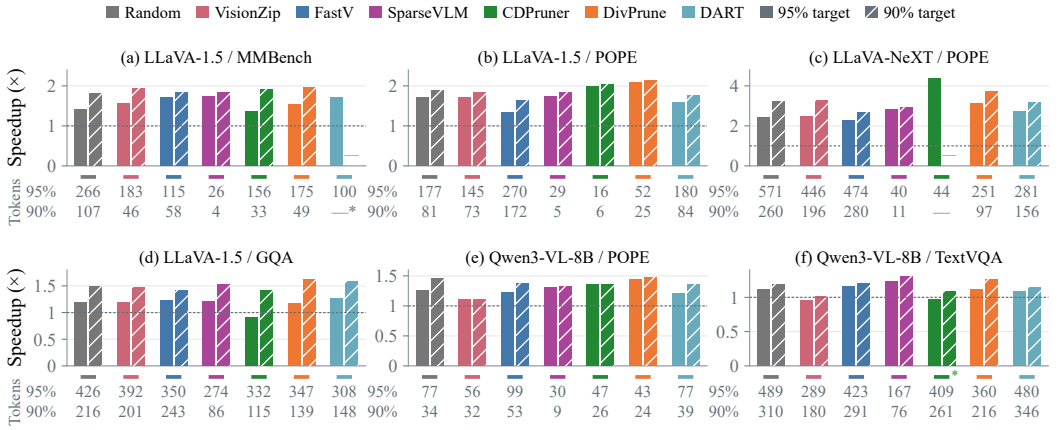


Figure 4: **Fewer tokens do not guarantee faster inference.** Bars show end-to-end speedup at nominal 95% and 90% dense-correct preservation targets; numbers below report retained visual tokens. Dashed lines mark the dense baseline (1×). Y-axis ranges vary.

time also depends on pruning location and selection overhead, making retained visual-token count insufficient to characterize the inference cost of the complete pruning procedure.

The correctness-preservation requirement can also change the speed ranking of methods. On LLaVA-1.5/MMBench, SparseVLM is faster than DivPrune at 95% preservation, with speedups of 1.74× and 1.53×, respectively. When the requirement is relaxed to 90%, DivPrune reaches 1.95× and surpasses SparseVLM's 1.83×. Comparisons of acceleration must therefore specify the correctness-preservation threshold. PCS reports the maximum measured speedup each method achieves among the evaluated configurations that satisfy a common threshold, thereby accounting jointly for token selection, pruning location, and algorithmic overhead.

6 Conclusion

Visual-token pruning should be evaluated by what it preserves, when token choice matters, and how much end-to-end acceleration it delivers in practical settings. PAA-Bench addresses these goals through dense-correct preservation, Random-Pass@k-based selection-difficulty stratification, and PCS under shared preservation targets. Across eight methods, three VLM backbones, and nine benchmarks, preservation-aware scoring changes method rankings, selection-sensitive examples expose clearer differences, and stronger token compression does not necessarily yield greater speedup. Efficiency rankings vary across tasks and preservation targets. Together, these findings support comparing pruning pipelines at matched preservation levels and reporting results across selection-difficulty strata, rather than relying on aggregate scores or retained-token counts alone.

AI USE STATEMENT

Generative AI was used to support the organization and development of the manuscript from a provided research outline, including drafting initial prose, restructuring sections, refining sentence flow, and improving clarity, coherence, and terminological consistency. It also assisted with preliminary reference discovery by identifying potentially relevant literature for further review.

REPRODUCIBILITY STATEMENT

The definitions of the proposed metrics and the complete evaluation protocol are provided in Section 4 and Appendix A. Dataset and benchmark coverage are described in Appendix B, while model configurations, token budgets, random interventions, and runtime measurement boundaries are documented in Appendix C. Complete token-selection and speedup results are reported in Appendices D and E, with qualitative records and sensitivity analyses provided in Appendices F and G.

REFERENCES

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

Jyoti Aneja, Michael Harrison, Neel Joshi, Tyler LaBonte, John Langford, and Eduardo Salinas. Phi-4-reasoning-vision-15b technical report. *arXiv preprint arXiv:2603.03975*, 2026.

Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. Agilepruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. *arXiv preprint arXiv:2603.01236*, 2026.

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.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3489–3499, June 2026.

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, 2024.

Jinhong Deng, Wen Li, Joey Tianyi Zhou, and Yang He. Scope: Saliency-coverage oriented token pruning for efficient multimodel llms. *Advances in Neural Information Processing Systems*, 38: 161527–161552, 2025.

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.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *IEEE International Conference on Computer Vision*, pp. 22826–22835, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, pp. 162549–162567, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

Lianyu Hu, Liqing Gao, Fanhua Shang, Liang Wan, and Wei Feng. illava: An image is worth fewer than 1/3 input tokens in large multimodal models. In *International Conference on Learning Representations*, volume 2026, pp. 73724–73737, 2026.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. In *AAAI Conference on Artificial Intelligence*, pp. 4075–4083, 2025.

Kaiyuan Li, Xiaoyue Chen, Chen Gao, Yong Li, and Xinlei Chen. Balanced token pruning: Accelerating vision language models beyond local optimization. *arXiv preprint arXiv:2505.22038*, 2025a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

Chenfei Liao, Wensong Wang, Zichen Wen, Xu Zheng, Yiyu Wang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Xin Zou, Yuqian Fu, Bin Ren, Linfeng Zhang, and Xuming Hu. Are we using the right benchmark: An evaluation framework for visual token compression methods. In *Annual Meeting of the Association for Computational Linguistics*, pp. 4236–4253, 2026.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233, 2024c.

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. In *Advances in Neural Information Processing Systems*, pp. 2507–2521, 2022.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Annual Meeting of the Association for Computational Linguistics*, pp. 20192–20204, 2025.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *IEEE International Conference on Computer Vision*, pp. 22857–22867, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Guinan Su, Li Shen, Lu Yin, Shiwei Liu, Yanwu Yang, and Jonas Geiping. GPTailor: Large language model pruning through layer cutting and stitching. In *International Conference on Learning Representations*, 2026.

Guohao Sun, Yufei Wang, Sizhuo Ma, Yuege Xie, Yuting Cheng, Zhiqiang Tao, and Jian Wang. If-prune: Information-flow guided token pruning for efficient vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3522–3531, June 2026.

Rinyoichi Takezoe, Yaqian Li, Zi-Hao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. Learn-Pruner: Rethinking attention-based token pruning in vision language models. In *International Conference on Learning Representations*, 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.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19769–19780, June 2025.

Hanshi Wang, Yuhao Xu, Zekun Xu, Jin Gao, Yufan Liu, Weiming Hu, Ke Wang, and Zhipeng Zhang. Each complexity deserves a pruning policy. *Advances in Neural Information Processing Systems*, 38:118209–118228, 2025a.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Zekun Wang, MingHua Ma, Zexin Wang, Rongchuan Mu, Liping Shan, Ming Liu, and Bing Qin. EffiVLM-BENCH: A comprehensive benchmark for evaluating training-free acceleration in large vision-language models. In *Annual Meeting of the Association for Computational Linguistics*, pp. 25546–25572, 2025b.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Conference on Empirical Methods in Natural Language Processing*, pp. 9961–9980, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14593–14603, June 2025.

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 *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025a.

Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23924–23934, October 2025b.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. In *IEEE International Conference on Computer Vision*, pp. 20857–20867, 2025a.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *Advances in Neural Information Processing Systems*, pp. 25438–25468, 2025b.

Shaolei Zhang, Qingkai Fang, Yang Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, volume 2025, pp. 53285–53310, 2025c.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pp. 74840–74857, 2025d.

Supplementary Material

This appendix fills in the evaluation details that are not developed in the main paper. Sections A–C clarify the metric definitions, benchmark coverage, and experimental protocol; Sections D and E report the complete token-selection and PCS results; Section F presents qualitative cases with their recorded random-pruning outcomes; and Section G examines how Random-Pass strata change across token budgets and how sensitive the assignments are to the number of random trials.

Contents

A Additional Details of the Evaluation Framework . . . 15
A.1 Correctness Preservation and Prediction Transitions . . . 15
A.2 Random-Pass Stratification and Its Interpretation . . . 15
A.3 Dataset and Stratum Aggregation . . . 16
A.4 From Measured Configurations to PCS . . . 16

B Datasets and Benchmark Coverage . . . 17
B.1 Coverage and Task Roles . . . 17
B.2 Natural-Image and Assistive Question Answering . . . 17
B.3 Text Reading and Science Questions . . . 17
B.4 Hallucination and Broad Multimodal Evaluation . . . 18

C Implementation and Evaluation Protocols . . . 19
C.1 Models, Methods, and Budget Coverage . . . 19
C.2 Dense References and Random Interventions . . . 19
C.3 Settings of the Main-Paper Diagnostics . . . 19
C.4 Runtime Boundaries and Configuration Selection . . . 20

D Complete Token-Selection Results and Analysis . . . 21
D.1 LLaVA-1.5-7B . . . 21
D.2 LLaVA-NeXT-7B . . . 22
D.3 Qwen3-VL-8B . . . 23
D.4 Qwen3-VL-2B . . . 24
D.5 LLaVA-1.5-13B . . . 25
D.6 Reading the Results across Model Sizes . . . 25

E Preservation-Constrained Speedup Results and Analysis . . . 26
E.1 LLaVA-1.5-7B . . . 26
E.2 LLaVA-NeXT-7B . . . 27
E.3 Qwen3-VL-8B . . . 28

F Qualitative Analysis of Random-Pruning Sensitivity . . . 29
F.1 Case Selection, Scoring, and Visual Encoding . . . 29
F.2 LLaVA-1.5-7B: Counting, Recognition, and Local Attributes . . . 30
F.3 LLaVA-NeXT-7B: Global and Local Token Views . . . 31
F.4 Qwen3-VL-8B: Relative Budgets and Answer Variability . . . 32

G Additional Analysis of Random-Pass Stratification . . . 33
G.1 Difficulty Distributions across Token Budgets . . . 33
G.2 Sensitivity to the Number of Random Trials . . . 35

A Additional Details of the Evaluation Framework

A.1 Correctness Preservation and Prediction Transitions

Aggregate accuracy records only the net change after pruning. Some predictions that the dense model gets right may become wrong, while some predictions that it gets wrong may become correct, and the two effects can offset each other. For a fixed model and dataset, let $b_i^D, b_i^P \in \{0, 1\}$ denote the correctness of the dense and pruned predictions under the same answer evaluator. Define $\mathcal{D}^+ = \{i : b_i^D = 1\}$ and $\mathcal{D}^- = \{i : b_i^D = 0\}$. Each example then falls into one of four transitions: correct-to-correct, correct-to-wrong, wrong-to-correct, or wrong-to-wrong. Over N examples,

$$\mathrm{Acc}_P - \mathrm{Acc}_D = \frac{N_{W\to C} - N_{C\to W}}{N}. \tag{7}$$

A small net accuracy drop can therefore hide many lost correct predictions when wrong-to-correct transitions offset them. To measure preservation directly, we condition on the dense model being correct:

$$\mathrm{Acc}_{\mathcal{D}^+}(P) = \frac{N_{C\to C}}{|\mathcal{D}^+|} = 1 - \frac{N_{C\to W}}{|\mathcal{D}^+|}, \qquad |\mathcal{D}^+| > 0. \tag{8}$$

PCS constrains this quantity. A preservation rate of 95% means that the pruned model keeps 95% of the predictions that the dense model originally answered correctly. It does not mean 95% accuracy on the full dataset, so preservation should be read alongside the dense model's task score.

Relation to relative accuracy. When both scores are binary accuracies on the same examples and $\mathrm{Acc}_D > 0$,

$$\frac{\mathrm{Acc}_P}{\mathrm{Acc}_D} = \mathrm{Acc}_{\mathcal{D}^+}(P) + \frac{N_{W\to C}}{|\mathcal{D}^+|}. \tag{9}$$

The second term is the contribution from wrong-to-correct transitions. Relative accuracy can therefore rise because pruning corrects dense-model errors, even when it does not preserve more of the dense model's correct predictions.

A.2 Random-Pass Stratification and Its Interpretation

Correctness preservation tells us how much survives pruning, but not whether the retained tokens were chosen well. To measure how much token choice matters for each example, we hold the model, token budget, pruning location, eligible positions, and decoding rule fixed, and vary only the random mask. Let p_i be the probability that a random mask preserves correctness on dense-correct example i. Under independent and identically distributed trials,

$$R_i = \text{Random-Pass@}\,k(x_i) \sim \mathrm{Binomial}(k, p_i), \quad \mathbb{E}[R_i/k] = p_i, \quad \mathrm{Var}(R_i/k) = \frac{p_i(1-p_i)}{k}. \tag{10}$$

Random-Pass@k records the *number of successful trials* among k random masks; it is not the probability of succeeding at least once. We use $k = 10$ and define Easy as 8–10 successes, Medium as 4–7, and Hard as 0–3. These labels describe sensitivity to random token selection under the specified intervention, not an intrinsic difficulty level of the original benchmark question.

Why stratification can expose selection gains. Suppose an informed selector succeeds on example i with probability $q_i \le 1$. Its gain over random selection satisfies $q_i - p_i \le 1 - p_i$. When p_i is already high, as it is for Easy examples, even a perfect selector has little room to improve over Random. Lower random success leaves more headroom. The wider method gaps in Figure 3 are an empirical result of the evaluated settings, not something implied by the definition alone.

A.3 Dataset and Stratum Aggregation

S-Avg and W-Avg answer different questions. S-Avg asks how a method performs when Easy, Medium, and Hard examples count equally, whereas W-Avg keeps the observed mixture of those strata within each dataset. For dataset d and stratum $s \in \{E, M, H\}$, let $n_{d,s}$ be the number of dense-correct examples and $a_{d,s}$ the fraction preserved by a method. We first compute

$$S_d = \frac{a_{d,E} + a_{d,M} + a_{d,H}}{3}, \qquad W_d = \frac{\sum_s n_{d,s} a_{d,s}}{\sum_s n_{d,s}}. \tag{11}$$

For J evaluated datasets, the reported percentage scores are

$$\text{S-Avg} = \frac{100}{J} \sum_{d=1}^{J} S_d, \qquad \text{W-Avg} = \frac{100}{J} \sum_{d=1}^{J} W_d. \tag{12}$$

Both metrics give every dataset the same weight. W-Avg weights examples within each dataset, so it is not an example-pooled average over all datasets. S-Avg gives the three strata equal weight, so it is not accuracy under the benchmark's original sample distribution. The individual dataset columns in the complete token-selection tables report $100S_d$.

A.4 From Measured Configurations to PCS

The fixed-budget tables ask how much correctness a method preserves at a particular token budget. PCS asks a different question: among the configurations that meet a shared preservation target, which measured configuration is fastest?

We first obtain dense predictions and correctness labels, use random pruning to form the difficulty strata, and evaluate each candidate pruning configuration on the same examples. PCS then selects the fastest measured configuration whose correctness preservation reaches the target. This comparison measures the inference cost of achieving the target instead of treating the final retained-token count as the cost itself.

For a fixed candidate set, lowering the preservation target can only enlarge the feasible set. Whenever at least one feasible configuration exists,

$$\tau_1 \le \tau_2 \quad \Longrightarrow \quad \text{PCS}_M@\tau_1 \ge \text{PCS}_M@\tau_2. \tag{13}$$

B Datasets and Benchmark Coverage

B.1 Coverage and Task Roles

The main token-selection tables cover nine benchmark variants for LLaVA-1.5-7B and eight each for LLaVA-NeXT-7B and Qwen3-VL-8B. The additional Qwen3-VL-2B and LLaVA-1.5-13B results cover seven variants and omit GQA and VQAv2. MMBench-EN and MMBench-CN are treated as separate variants, while VQAv2 appears only in the LLaVA-1.5-7B table. The benchmark set spans natural-image reasoning, text reading, science questions, object hallucination, broad perception, and questions collected in an assistive-use setting. Table 3 summarizes the exact coverage used in the complete token-selection results.

Table 3: Benchmark variants in the complete token-selection tables. "Primary three" denotes LLaVA-1.5-7B, LLaVA-NeXT-7B, and Qwen3-VL-8B; "All five" additionally includes Qwen3-VL-2B and LLaVA-1.5-13B. Coverage here refers to the reported token-selection results, not every PCS or diagnostic experiment.

Benchmark / table label	Main capability examined	Model coverage
GQA	Objects, attributes, relations, and compositional reasoning	Primary three
VizWiz	Real-world visual questions from blind users	All five
ScienceQA-IMG / SQA^I	Science questions with image context	All five
TextVQA / VQA^T	Reading and reasoning about text in images	All five
POPE	Object existence and object hallucination	All five
MME	Perceptual and cognitive judgments	All five
VQAv2 / VQA^{v2}	Open-ended natural-image question answering	LLaVA-1.5-7B
MMBench-EN / MMB	General multimodal perception and reasoning	All five
MMBench-CN / MMB^{CN}	Chinese-language multimodal evaluation	All five

B.2 Natural-Image and Assistive Question Answering

GQA. GQA (Hudson & Manning, 2019) builds compositional questions from scene graphs, including attributes, object relations, and multi-step reasoning. These questions are useful for token pruning because the answer may depend on several objects or on a relation between them; preserving only the most salient object may not be enough. The GQA column in our tables reports stratified correctness preservation, not the benchmark's original full-set accuracy.

VQAv2. VQAv2 (Goyal et al., 2017) pairs open-ended questions with natural images and multiple human answers. Its complementary image pairs reduce the value of answering from the question alone, adding a broad natural-image task to the LLaVA-1.5-7B comparison. The native evaluator uses answer normalization and consensus-based fractional credit, whereas the transition analysis requires a binary correct/incorrect event. We keep those two quantities separate.

VizWiz. VizWiz (Gurari et al., 2018) contains user-captured images and spoken questions from blind users, together with multiple human answers and an explicit unanswerable category. It tests whether pruning preserves useful evidence in imperfect, real-world images without turning insufficient evidence into an overconfident answer. Preservation scores on this dataset therefore depend on how answer normalization and unanswerable cases are handled; the task is not simply central-object recognition.

B.3 Text Reading and Science Questions

TextVQA. TextVQA (Singh et al., 2019) requires the model to read and reason about text in an image. A selector can preserve the broad scene and still remove a small sign or label that determines the answer, making this task particularly sensitive to fine visual evidence. Whether OCR text is

supplied in the prompt also matters, because external text can reduce the model's dependence on the retained visual tokens.

ScienceQA-IMG. ScienceQA (Lu et al., 2022) contains multiple-choice science questions with image and/or text context; our tables use the image subset, denoted SQA^I. The task combines visual interpretation with scientific knowledge, but the presence of an image does not mean that every question requires it. Random-Pass strata add that missing information by measuring how strongly the evaluated model's answer depends on visual-token selection.

B.4 Hallucination and Broad Multimodal Evaluation

POPE. POPE (Li et al., 2023) asks whether particular objects are present in an image and is designed to probe object hallucination. Its random, popular, and adversarial variants differ in how absent objects are sampled. For token pruning, POPE tests whether the evidence needed for a yes/no object judgment remains available after selection. A high POPE preservation score should not be read as evidence that fine-grained reading or relational reasoning is also preserved; those abilities are covered by other tasks.

MME. MME (Fu et al., 2025) evaluates perception and cognition across 14 subtasks using short instruction–answer pairs. Its breadth complements the more focused benchmarks. The percentage in our token-selection tables is the stratum-balanced preservation score defined in Section A.3, not the original MME composite score. Reproducing that value requires the same subtask set and the same unit for assigning binary correctness labels.

MMBench-EN and MMBench-CN. MMBench (Liu et al., 2024c) evaluates multimodal perception and reasoning with multiple-choice questions. We report the English and Chinese variants separately to show how methods behave across the two language settings. Differences between the columns cannot be attributed to visual selection alone, however; language understanding, answer extraction, and evaluator version can also affect them.

C Implementation and Evaluation Protocols

C.1 Models, Methods, and Budget Coverage

The primary experiments use three backbones: LLaVA-1.5-7B, LLaVA-NeXT-7B, and Qwen3-VL-8B. The supplementary token-selection results add Qwen3-VL-2B and LLaVA-1.5-13B, without runtime measurements for either configuration. As in the main paper, all runs use a PRO 6000 GPU and batch size 1; methods within a comparison share the same prompts and a maximum output length of 128 tokens.

The complete LLaVA-1.5-7B table includes CDPruner, DivPrune, FastV, VisionZip, DART, PruMerge, SparseVLM, and VisPruner (Zhang et al., 2025a), plus Random. LLaVA-NeXT-7B includes the same set except VisPruner. The Qwen3-VL-8B selection table includes CDPruner, DivPrune, FastV, VisionZip, and DART, plus Random. Both additional model tables also report these five methods and Random.

Table 4: Budgets and coverage of the complete fixed-budget selection tables. Qwen3-VL uses retention fractions rather than a common absolute token count.

Model	Budgets	Benchmark variants	Methods + Random
LLaVA-1.5-7B	128, 64, 32 tokens	9	$8+1$
LLaVA-NeXT-7B	640, 320, 160 tokens	8	$7+1$
Qwen3-VL-8B	10%, 20% retention	8	$5+1$
Qwen3-VL-2B	20%, 10% retention	7	$5+1$
LLaVA-1.5-13B	128, 64, 32 tokens	7	$5+1$

C.2 Dense References and Random Interventions

Correctness preservation is a paired comparison. Each pruned prediction must be matched to the dense prediction for the same example, using the same prompt, image processing, answer evaluator, and decoding rule. Changing any of these choices can change $\mathcal{D}^+$ and therefore the denominator of the preservation score. The random interventions use the same evaluation setup; only the token mask changes across trials.

C.3 Settings of the Main-Paper Diagnostics

Difficulty-stratified comparison. Figure 3 forms the Easy, Medium, and Hard strata separately for each model and token budget. For Qwen3-VL, the figure uses adapted implementations of CDPruner, FastV, VisionZip, and DART; SparseVLM appears only in the LLaVA-1.5 panels. Bar labels are rounded to integer percentages, while the bar heights use the unrounded values.

Transitions and correction overlap. Figure 2(a) uses 64 retained tokens for LLaVA-1.5 and 20% retention for Qwen3-VL-8B. Panel (b) averages the LLaVA-1.5 results equally across GQA, POPE, and MMBench-EN. A stable correction is a dense-incorrect example that becomes correct in all five random-pruning trials; an occasional correction succeeds in one to four trials. These five-trial correction groups are separate from the ten-trial difficulty strata defined on $\mathcal{D}^+$. Panel (c) uses 128 tokens for LLaVA-1.5, 320 for LLaVA-NeXT, and 20% retention for Qwen3-VL-8B.

For a dense-incorrect example, we estimate the correction frequency as

$$\widehat{p}_i^{\,\mathrm{corr}} = \frac{1}{5}\sum_{j=1}^{5} b_i^{(j)}, \qquad i \in \mathcal{D}^-. \tag{14}$$

Text-only overlap is the fraction of examples in a correction group that are also answered correctly without image input. A higher overlap is consistent with greater reliance on textual knowledge, but it does not identify the cause of every correction. The control also removes the image before multimodal fusion, which is not the same intervention as deleting visual tokens after image information has already entered other representations.

C.4 Runtime Boundaries and Configuration Selection

Retained-token count captures only part of the cost of a pruning method. End-to-end timing must cover the same boundary for the dense and pruned systems, including the vision encoder, projector, language-model execution, and selection work such as scoring and token gathering. CPU preprocessing and device transfer should either be included for both systems or excluded for both and reported separately.

This matched boundary is especially important when pruning happens after several decoder layers: a method may retain few tokens at the end while still paying for substantial computation before pruning. PCS therefore uses measured pipeline time, including selection overhead, rather than inferring efficiency from the final token count.

D Complete Token-Selection Results and Analysis

Tables 5–9 show the fixed-budget token-selection results, aggregated as described in Section A.3. All scores are percentages, and all differences are reported in percentage points.

D.1 LLaVA-1.5-7B

Table 5: LLaVA-1.5 results on nine benchmarks with 128, 64, and 32 retained visual tokens. S-Avg and W-Avg are equal-weight averages across datasets of the three-stratum and sample-weighted scores, respectively. Bold and underlined values indicate the best and second-best results, respectively, within each column and token budget.

Method	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	VQA^{v2}	MMB	MMB^{CN}	S-Avg	W-Avg
Retain 128 visual tokens											
Random	56.71	56.33	55.22	55.11	57.79	56.48	57.34	55.28	54.34	56.07	85.61
CDPruner	73.75	74.38	71.68	86.74	68.89	73.69	80.01	69.16	70.09	74.27	92.10
DivPrune	68.80	72.74	70.11	83.31	68.68	69.48	74.86	66.80	66.18	71.22	91.21
FastV	63.29	81.11	**85.23**	89.65	73.10	**83.15**	75.44	77.17	82.13	78.92	90.94
VisionZip	65.27	78.96	76.82	88.60	68.41	73.85	76.68	65.70	76.21	74.50	91.99
DART	72.75	83.79	85.03	86.08	79.69	81.34	80.88	**79.64**	83.25	81.38	92.96
PruMerge	61.82	73.25	71.02	80.77	69.89	73.38	74.28	65.94	72.03	71.38	90.10
SparseVLM	**74.39**	**84.07**	81.62	**92.26**	**79.76**	78.92	**83.64**	78.18	**85.63**	**82.05**	**94.43**
VisPruner	65.98	78.85	75.60	89.61	71.89	74.98	76.87	70.29	71.34	75.05	92.25
Retain 64 visual tokens											
Random	56.15	55.56	54.61	53.48	58.24	55.50	56.81	55.52	54.85	55.64	79.94
CDPruner	73.93	74.15	72.12	84.32	79.94	75.34	82.52	75.11	71.98	76.60	89.33
DivPrune	69.16	72.01	70.69	82.36	79.08	69.38	76.48	66.02	68.40	72.62	88.02
FastV	55.12	72.31	81.92	82.16	58.73	71.53	65.61	72.18	74.56	70.46	83.13
VisionZip	64.33	77.83	80.63	87.28	67.73	73.70	75.50	69.66	76.45	74.79	88.28
DART	66.62	77.35	80.29	79.56	71.53	80.17	75.51	75.41	78.14	76.06	87.76
PruMerge	61.62	74.19	77.14	83.27	67.65	74.05	74.09	68.88	74.91	72.87	86.74
SparseVLM	**76.51**	**84.12**	**86.11**	**91.96**	**82.68**	**83.01**	**86.81**	**86.05**	**90.12**	**85.26**	**93.26**
VisPruner	65.09	78.01	79.12	87.50	76.99	72.96	75.86	68.57	74.42	75.39	88.61
Retain 32 visual tokens											
Random	55.86	54.40	54.31	52.87	57.23	55.16	56.09	54.56	54.29	54.97	73.80
CDPruner	74.43	73.52	74.17	80.65	82.86	73.99	82.92	75.00	67.41	76.11	85.46
DivPrune	67.54	70.52	71.81	78.18	78.02	70.57	75.64	69.76	71.48	72.61	83.90
FastV	50.54	58.26	76.67	70.51	51.05	61.92	54.83	63.60	64.89	61.36	72.54
VisionZip	59.52	73.02	78.87	81.47	67.24	73.41	70.63	71.27	71.83	71.92	82.27
DART	61.68	68.98	77.38	71.67	68.79	73.04	68.61	70.71	72.06	70.32	80.71
PruMerge	60.00	73.68	77.13	80.99	62.37	74.49	70.37	68.58	72.98	71.18	81.92
SparseVLM	**79.20**	**82.54**	**85.69**	**89.38**	**86.50**	**87.40**	**88.17**	**88.22**	**93.60**	**86.74**	**91.63**
VisPruner	61.37	71.97	75.26	82.62	71.62	74.01	71.31	71.06	68.54	71.97	82.68

SparseVLM holds up best as the budget shrinks. SparseVLM has the highest S-Avg and W-Avg at all three budgets. Reducing the budget from 128 to 32 tokens lowers its W-Avg by only 2.80 points, from 94.43 to 91.63. Over the same range, FastV drops by 18.40 points (90.94 to 72.54), and CDPruner drops by 6.64 points (92.10 to 85.46). The methods therefore spread farther apart as compression becomes more aggressive. At 32 tokens, FastV's W-Avg of 72.54 even falls below Random's 73.80.

The aggregate does not tell the whole story. At 128 tokens, FastV has the highest score on ScienceQA-IMG and MME, while DART leads on MMBench-EN. A method can therefore lead an aggregate without leading every task. The large gap between Random's W-Avg of 85.61 and S-Avg of 56.07 also shows how strongly the sample mix shapes the final score.

D.2 LLaVA-NeXT-7B

Table 6: Complete LLaVA-NeXT results with 640, 320, and 160 retained visual tokens. S-Avg denotes the three-stratum aggregate and W-Avg the sample-weighted aggregate. Bold and underlined values indicate the best and second-best results, respectively, within each column and token budget.

Method	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	MMB	MMB^{CN}	S-Avg	W-Avg
Retain 640 visual tokens										
Random	57.16	55.96	56.88	55.29	59.06	57.99	56.31	56.29	56.87	89.57
CDPruner	65.81	73.74	67.49	82.59	77.47	76.54	80.35	69.28	74.16	92.87
DivPrune	62.49	69.79	68.69	75.07	74.96	75.06	67.93	58.94	69.12	91.71
FastV	79.78	84.88	84.40	**91.48**	79.95	85.02	80.84	87.45	84.22	95.76
VisionZip	64.67	75.05	63.82	86.04	73.53	72.45	73.22	66.29	71.88	92.73
DART	**83.21**	**86.82**	**86.04**	90.25	**89.12**	**86.33**	80.67	**87.58**	**86.25**	**96.13**
PruMerge	60.24	66.15	59.34	71.29	72.59	65.52	50.39	57.54	62.88	89.77
SparseVLM	76.30	82.99	75.73	90.71	82.17	77.17	**84.02**	84.44	81.69	95.84
Retain 320 visual tokens										
Random	56.63	54.95	56.17	53.65	58.26	56.16	57.07	55.21	56.01	84.78
CDPruner	69.90	72.49	60.90	81.73	86.47	72.73	76.37	63.10	72.96	89.82
DivPrune	63.78	69.40	62.13	74.50	80.42	70.57	73.65	64.64	69.89	88.91
FastV	72.23	77.69	**79.35**	86.40	71.01	76.12	75.91	81.31	77.50	90.84
VisionZip	63.52	75.56	61.22	85.03	75.11	68.42	70.35	63.87	70.38	88.95
DART	77.40	82.08	77.61	85.38	81.74	**82.61**	79.12	85.91	81.48	92.32
PruMerge	62.12	69.70	58.85	78.23	76.18	68.13	61.39	58.77	66.67	87.14
SparseVLM	**78.41**	**83.60**	72.57	**91.62**	**89.85**	81.69	**90.07**	**88.01**	**84.48**	**94.56**
Retain 160 visual tokens										
Random	56.61	54.49	56.65	52.88	57.34	56.58	55.44	54.70	55.59	79.70
CDPruner	72.30	71.74	65.88	80.95	90.56	73.16	72.20	67.79	74.32	87.68
DivPrune	64.93	70.96	67.26	75.06	81.20	67.40	71.60	67.61	70.75	86.28
FastV	60.40	61.69	**72.79**	74.03	55.82	67.83	68.36	69.87	66.35	80.69
VisionZip	60.39	72.26	67.10	81.53	68.88	61.24	66.18	61.47	67.38	83.40
DART	69.24	76.32	71.75	77.28	74.49	76.20	71.16	71.82	73.53	85.93
PruMerge	61.28	68.12	64.76	79.99	73.09	68.72	59.87	60.81	67.08	83.08
SparseVLM	**80.83**	**83.85**	72.65	**90.95**	**93.29**	**86.13**	**85.93**	**87.71**	**85.17**	**93.18**

The ranking changes as the budget tightens. At 640 tokens, DART leads both aggregates with an S-Avg/W-Avg of 86.25/96.13. SparseVLM takes the lead at 320 and 160 tokens, reaching 85.17/93.18 at the smallest budget. From 640 to 160 tokens, FastV's W-Avg falls by 15.07 points (95.76 to 80.69), DART's by 10.20 points (96.13 to 85.93), and SparseVLM's by only 2.66 points (95.84 to 93.18). A ranking obtained at a generous budget does not carry over to stronger compression.

No single aggregate captures what happens on every task. At 160 tokens, SparseVLM leads seven of the eight dataset columns. The only exception is ScienceQA-IMG, where FastV is slightly higher at 72.79 versus 72.65. CDPruner also varies sharply across tasks, scoring 90.56 on POPE but only 65.88 on ScienceQA-IMG. These differences are easy to miss in a single summary number, so the task-level results remain essential.

S-Avg makes differences in less common difficulty groups more visible. At 640 tokens, DART is 6.56 points above Random in W-Avg but 29.38 points above it in S-Avg. At 160 tokens, SparseVLM's corresponding margins are 13.48 and 29.58 points. The reason is straightforward: S-Avg gives Easy, Medium, and Hard examples equal weight, so groups with fewer samples have more influence than they do under W-Avg.

D.3 QWEN3-VL-8B

Table 7: Qwen3-VL results at 10% and 20% visual-token retention. S-Avg denotes the three-stratum aggregate and W-Avg the sample-weighted aggregate. Bold and underlined values indicate the best and second-best results, respectively, within each column and retention budget.

Method	GQA	VizWiz	SQA^I	VQA^T	POPE	MME	MMB	MMB^{CN}	S-Avg	W-Avg
Retain 10% of visual tokens										
Random	56.37	55.67	55.08	54.32	56.60	56.47	56.44	56.58	55.94	79.14
CDPruner	<u>60.35</u>	59.65	59.31	<u>69.35</u>	59.79	60.20	**64.31**	**62.14**	61.89	<u>81.62</u>
DivPrune	**65.50**	<u>60.55</u>	56.87	**70.79**	78.22	**63.69**	<u>61.02</u>	<u>57.37</u>	**64.25**	**82.19**
FastV	54.60	50.03	51.83	52.53	<u>83.59</u>	<u>60.41</u>	55.16	53.64	57.72	72.36
VisionZip	58.23	**69.53**	<u>62.52</u>	61.23	**85.37**	60.37	57.09	54.28	<u>63.58</u>	80.08
DART	55.62	53.57	**66.25**	50.10	66.58	56.33	54.95	57.02	57.55	76.53
Retain 20% of visual tokens										
Random	56.96	56.50	55.64	55.86	58.16	57.47	58.07	57.06	56.97	86.59
CDPruner	63.46	60.71	63.79	71.47	75.67	**67.21**	**66.79**	<u>65.06</u>	<u>66.77</u>	88.53
DivPrune	**71.02**	61.66	58.59	<u>74.17</u>	84.21	63.76	60.68	58.14	66.53	<u>88.95</u>
FastV	60.73	<u>61.96</u>	56.25	64.04	<u>84.79</u>	61.84	58.07	60.24	63.49	83.29
VisionZip	<u>66.20</u>	**75.10**	<u>65.36</u>	**75.27**	**88.61**	<u>66.21</u>	<u>64.98</u>	62.93	**70.58**	**89.64**
DART	59.62	61.90	**67.03**	54.01	70.70	60.15	63.98	**66.30**	62.96	84.72

The preferred selector depends on the retention rate. At 10% retention, DivPrune has the highest S-Avg/W-Avg at 64.25/82.19. At 20%, VisionZip leads with 70.58/89.64. Even at the same 20% retention rate, CDPruner and DivPrune switch order depending on the aggregate: CDPruner is slightly higher in S-Avg (66.77 versus 66.53), whereas DivPrune is higher in W-Avg (88.95 versus 88.53).

A strong result on one task can hide weak results elsewhere. With 10% retention, FastV scores 83.59 on POPE but only 50.03 on VizWiz and 52.53 on TextVQA. Its S-Avg is 1.78 points above Random, yet its W-Avg of 72.36 is 6.78 points below Random. In other words, the same selector can look better or worse depending on how the difficulty groups are weighted. Both aggregates and the per-dataset results are needed before treating an improvement as broad.

A strong Random baseline still leaves room for better selection. At 20% retention, Random reaches a W-Avg of 86.59 but an S-Avg of only 56.97. VisionZip raises these scores to 89.64 and 70.58, gains of 3.05 and 13.61 points, respectively. The much larger gain in S-Avg is consistent with the main-paper finding that Easy examples can make improvements look smaller in an aggregate dominated by the observed sample mix.

D.4 QWEN3-VL-2B

Table 8: Token-selection results for **Qwen3-VL-2B** on seven benchmarks. Dataset columns report three-stratum preservation scores; S-Avg and W-Avg follow Section A.3. All values are percentages. Bold and underlined values mark the best and second-best scores within each column and budget, respectively.

Method	VizWiz	SQA^I	TextVQA	POPE	MME	MMB-EN	MMB-CN	S-Avg	W-Avg
Retain 20% of visual tokens									
Random	56.64	56.45	55.16	54.72	56.51	56.45	57.24	56.17	86.27
CDPruner	67.43	**74.81**	<u>82.71</u>	**100.00**	**70.51**	<u>73.08</u>	**68.56**	**76.73**	<u>91.53</u>
DivPrune	**68.98**	<u>71.67</u>	**83.34**	66.67	<u>69.58</u>	**75.73**	<u>66.73</u>	<u>71.82</u>	**92.34**
FastV	<u>67.75</u>	49.57	72.99	44.44	61.38	53.32	53.12	57.51	83.75
VisionZip	65.75	61.91	57.46	<u>77.78</u>	56.27	57.23	56.70	61.87	85.71
DART	60.25	65.75	63.09	41.66	62.18	64.18	65.82	60.42	86.64
Retain 10% of visual tokens									
Random	55.56	55.30	53.31	<u>58.78</u>	55.56	56.53	55.75	55.83	79.27
CDPruner	<u>67.42</u>	**73.69**	<u>78.47</u>	**73.33**	**68.95**	**69.42**	65.10	**70.91**	<u>86.56</u>
DivPrune	**68.34**	<u>70.08</u>	**79.12**	55.56	<u>65.10</u>	<u>69.37</u>	**65.58**	<u>67.59</u>	**86.86**
FastV	43.69	44.89	51.10	44.44	50.91	42.80	40.95	45.54	69.27
VisionZip	54.53	52.91	44.10	56.67	52.82	53.20	50.55	52.11	75.85
DART	53.02	63.29	60.71	44.44	61.94	65.17	<u>65.16</u>	59.11	80.22

S-Avg and W-Avg favor different methods. CDPruner has the highest S-Avg at both 20% and 10% retention, with scores of 76.73 and 70.91. DivPrune instead has the highest W-Avg, at 92.34 and 86.86. At 20%, CDPruner is 4.91 points ahead of DivPrune in S-Avg but 0.81 points behind in W-Avg; at 10%, the corresponding gaps are 3.32 and 0.30 points. Which method leads therefore depends on whether each difficulty group receives equal weight or keeps its observed share of the data.

FastV deteriorates sharply under stronger compression. When retention drops from 20% to 10%, FastV's W-Avg falls from 83.75 to 69.27, whereas Random falls from 86.27 to 79.27. At the smaller budget, FastV is therefore 10.00 points below Random. VisionZip also trails Random at both budgets, while CDPruner and DivPrune remain above Random in both aggregates.

A task score only makes sense with its evaluation denominator. CDPruner's POPE score of 100.00 at 20% retention is the three-stratum preservation score computed on examples that the dense model originally answered correctly. It is not 100% accuracy on the full POPE benchmark. The number of examples in each stratum is also needed to judge the precision and coverage of this result. In addition, this seven-task aggregate excludes GQA and VQAv2, so it should not be compared directly with the aggregate in the Qwen3-VL-8B table.

D.5 LLaVA-1.5-13B

Table 9: Token-selection results for **LLaVA-1.5-13B** on seven benchmarks. Dataset columns report three-stratum preservation scores; S-Avg and W-Avg follow Section A.3. All values are percentages. Bold and underlined values mark the best and second-best scores within each column and budget, respectively.

Method	VizWiz	SQA^I	TextVQA	POPE	MME	MMB-EN	MMB-CN	S-Avg	W-Avg
Retain 128 visual tokens									
Random	56.44	58.24	54.60	57.23	58.04	56.39	55.00	56.56	86.34
CDPruner	73.72	73.08	85.44	66.80	69.79	71.35	67.20	72.48	92.44
DivPrune	74.54	68.76	81.40	64.78	66.65	67.71	63.49	69.62	91.88
FastV	**82.73**	**88.90**	86.38	79.88	**89.36**	**84.87**	78.63	**84.39**	93.95
VisionZip	78.82	78.40	**88.11**	68.05	70.17	69.59	67.77	74.42	93.07
DART	82.60	88.89	85.98	**82.46**	86.56	79.32	**84.70**	84.36	**94.25**
Retain 64 visual tokens									
Random	55.26	55.00	53.46	57.41	57.87	55.91	55.26	55.74	81.28
CDPruner	75.16	72.21	84.11	73.17	71.47	75.36	71.34	74.69	90.13
DivPrune	74.01	66.07	79.71	70.59	71.91	70.10	67.72	71.44	88.98
FastV	74.73	**83.20**	78.31	73.76	78.80	76.00	79.80	77.80	88.04
VisionZip	**78.16**	80.14	**86.59**	69.05	72.38	74.20	72.07	76.09	**90.37**
DART	73.72	80.26	75.98	**76.25**	**80.00**	**82.09**	**81.79**	**78.59**	89.01
Retain 32 visual tokens									
Random	54.17	55.38	52.66	56.52	56.59	55.36	55.14	55.12	75.32
CDPruner	73.83	74.59	78.11	**78.69**	74.75	**74.07**	71.97	**75.15**	**86.26**
DivPrune	71.80	71.58	76.05	74.93	74.14	66.92	73.50	72.70	85.27
FastV	63.41	**81.40**	67.46	69.46	69.54	66.88	70.60	69.82	79.84
VisionZip	**74.34**	77.85	**81.42**	67.30	67.87	71.77	74.35	73.56	85.07
DART	63.10	79.28	68.01	73.93	**75.73**	71.38	**74.37**	72.26	81.79

The leader changes with both the budget and the aggregate. At 128 tokens, FastV has the highest S-Avg at 84.39, only 0.03 points above DART, while DART has the highest W-Avg at 94.25. At 64 tokens, DART leads S-Avg (78.59) and VisionZip leads W-Avg (90.37). At 32 tokens, CDPruner leads both aggregates, with scores of 75.15 and 86.26. The ordering at a larger budget therefore does not tell us what the ordering will be under stronger compression.

Some task-specific strengths persist across budgets. VisionZip has the highest TextVQA score at all three budgets—88.11, 86.59, and 81.42—even when another method leads the aggregate. From 128 to 32 tokens, CDPruner's W-Avg drops by 6.18 points, compared with 14.11 points for FastV and 12.46 points for DART. This separates two questions that an overall score can blur: which method is strongest on a particular task, and which method remains stable as the token budget shrinks. As with the other backbones, changes in S-Avg also partly reflect the fact that the difficulty strata themselves change with the budget.

D.6 Reading the Results across Model Sizes

Taken together, the five tables show that method rankings change with the budget, the task, and the aggregation rule. Fixed-budget preservation also says nothing about realized efficiency. The next section therefore reports measured speedups only for the three primary model configurations.

E PRESERVATION-CONSTRAINED SPEEDUP RESULTS AND ANALYSIS

The following three tables report the updated operating points for three model backbones on MMBench-EN, TextVQA, POPE, and GQA. Each backbone has a separate table, with results shown at 95% and 90% correctness preservation. These percentages refer to the fraction of predictions that were correct under the dense model and remain correct after pruning; they do not indicate the fraction of visual tokens retained. Speedup is measured relative to the corresponding dense-model baseline, while the actual number of retained tokens is reported separately.

E.1 LLAVA-1.5-7B

Table 10: End-to-end speedup and retained visual tokens for **LLaVA-1.5-7B**. Bold marks the highest reported speedup within each dataset and preservation target, including ties and Random. Dashes denote unreported results.

Method	**MMBench-EN**		**TextVQA**		**POPE**		**GQA**	
	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens
95% correctness-preservation target								
Dense	1.00×	576	1.00×	576	1.00×	576	1.00×	576
Random	1.41×	266	1.08×	461	1.71×	177	1.19×	426
VisionZip	1.56×	183	1.32×	230	1.70×	145	1.18×	392
FastV	1.71×	115	**1.41**×	142	1.33×	270	1.23×	350
SparseVLM	**1.74**×	26	1.33×	116	1.74×	29	1.22×	274
CDPruner	1.35×	156	1.04×	262	1.98×	16	0.90×	332
DivPrune	1.53×	175	1.20×	297	**2.08**×	52	1.18×	347
DART	1.72×	100	1.33×	180	1.59×	180	**1.26**×	308
90% correctness-preservation target								
Dense	1.00×	576	1.00×	576	1.00×	576	1.00×	576
Random	1.82×	107	1.24×	312	1.89×	81	1.49×	216
VisionZip	1.94×	46	**1.53**×	85	1.84×	73	1.46×	201
FastV	1.83×	58	1.51×	62	1.64×	172	1.42×	243
SparseVLM	1.83×	4	1.45×	23	1.84×	5	1.52×	86
CDPruner	1.91×	33	1.35×	94	2.03×	6	1.42×	115
DivPrune	**1.95**×	49	**1.53**×	85	**2.13**×	25	**1.61**×	139
DART	—		1.47×	76	1.75×	84	1.58×	148

The leading method depends on the task and preservation target. At 95% correctness preservation, SparseVLM achieves the highest speedup on MMBench-EN (1.74×), FastV on TextVQA (1.41×), DivPrune on POPE (2.08×), and DART on GQA (1.26×). When the preservation target is relaxed to 90%, DivPrune reaches the highest speedup on all four tasks, tying VisionZip at 1.53× on TextVQA. Its GQA speedup also increases from 1.18× to 1.61×, surpassing DART's 1.58×. Thus, the ordering of methods at one preservation target does not determine their ordering at another.

Fewer tokens do not necessarily yield greater speedup. On MMBench-EN at the 90% preservation target, SparseVLM retains only 4 tokens and reaches a $1.853times$ speedup, whereas DivPrune retains 49 tokens but is faster at 1.95×. On GQA at 95% preservation, CDPruner retains 332 of the dense model's 576 tokens yet achieves only 0.90×, meaning that pruning causes a net slowdown at this operating point. The final token count does not capture computation already performed before pruning, nor does it reflect the extra overhead introduced by token selection itself.

E.2 LLaVA-NeXT-7B

Table 11: End-to-end speedup and retained visual tokens for **LLaVA-NeXT-7B**. Bold marks the highest reported speedup within each dataset and preservation target, including ties and Random. Dashes denote unreported results.

Method	MMBench-EN		TextVQA		POPE		GQA	
	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens
95% correctness-preservation target								
Dense	1.00×	1927	1.00×	2310	1.00×	2250	1.00×	2239
Random	2.07×	617	1.12×	1951	2.43×	571	1.35×	1482
VisionZip	1.94×	626	1.07×	2022	2.47×	446	—	
FastV	2.24×	385	**1.84**×	577	2.29×	474	1.73×	896
SparseVLM	**2.53**×	65	1.59×	623	2.80×	40	1.54×	906
CDPruner	2.13×	474	—		**4.37**×	44	—	
DivPrune	2.01×	525	0.81×	2199	3.11×	251	—	
DART	2.35×	355	1.81×	579	2.71×	281	**1.82**×	783
90% correctness-preservation target								
Dense	1.00×	1927	1.00×	2310	1.00×	2250	1.00×	2239
Random	3.31×	174	1.30×	1478	3.20×	260	2.25×	653
VisionZip	3.16×	157	2.08×	468	3.26×	196	1.80×	891
FastV	2.75×	177	2.15×	276	2.67×	280	2.29×	448
SparseVLM	2.79×	9	1.96×	138	2.92×	11	2.23×	233
CDPruner	**3.76**×	45	1.69×	717	**4.62**×	10	2.22×	558
DivPrune	3.68×	76	1.27×	1155	3.74×	97	1.87×	705
DART	2.93×	140	**2.17**×	288	3.15×	156	**2.48**×	371

The attainable speedup remains task dependent. At 95% correctness preservation, the highest reported speedup ranges from DART's $1.82\times$ on GQA to CDPruner's $4.37\times$ on POPE. SparseVLM is fastest on MMBench-EN at $2.53\times$, while FastV leads on TextVQA at $1.84\times$. On POPE, CDPruner retains 44 tokens and is still faster than SparseVLM, which retains 40 tokens and reaches $2.80\times$. This again shows that similar final token budgets can lead to different realized runtime costs across methods.

Relaxing the preservation target to 90% benefits methods differently. CDPruner reaches $3.76\times$ on MMBench-EN and $4.62\times$ on POPE, while DART leads on TextVQA and GQA at $2.17\times$ and $2.48\times$, respectively. CDPruner's newly reported POPE operating point retains only 10 tokens and improves further over its $4.37\times$ result at 95% preservation. On TextVQA, DivPrune improves from $0.81\times$ at 95% preservation to $1.27\times$ at 90%. This also shows that satisfying a correctness-preservation constraint does not necessarily translate into a net speedup.

Task-level differences matter when comparing speedups. The dense-model token count is not the same across tasks: MMBench-EN starts from 1,927 tokens, while TextVQA starts from 2,310. The four tasks therefore do not have identical initial workloads. The relatively modest speedups on TextVQA are consistent with the task's sensitivity to fine-grained visual information, but this table alone cannot separate that effect from image resolution, generation length, pruning location, or implementation details. At 90% preservation, the gap between DART and FastV on TextVQA is also small ($2.17\times$ versus $2.15\times$).

E.3 QWEN3-VL-8B

Table 12: End-to-end speedup and retained visual tokens for **Qwen3-VL-8B**. Bold marks the highest reported speedup within each dataset and preservation target, including ties and Random. Dashes denote unreported results.

Method	MMBench-EN		TextVQA		POPE		GQA	
	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens	Speedup	Tokens
95% correctness-preservation target								
Dense	1.00×	155	1.00×	750	1.00×	269	1.00×	268
Random	1.12×	45	1.11×	489	1.26×	77	1.17×	186
VisionZip	1.02×	45	0.95×	289	1.10×	56	0.97×	141
FastV	1.03×	65	1.15×	423	1.22×	99	1.16×	179
SparseVLM	1.12×	17	**1.23**×	167	1.31×	30	1.17×	109
CDPruner	1.13×	45	0.97×	409	1.35×	47	1.03×	165
DivPrune	**1.15**×	45	1.11×	360	**1.47**×	43	**1.18**×	151
DART	1.10×	48	1.09×	480	1.21×	77	1.14×	153
90% correctness-preservation target								
Dense	1.00×	155	1.00×	750	1.00×	269	1.00×	268
Random	1.13×	24	1.19×	310	**1.45**×	34	1.23×	107
VisionZip	1.03×	25	1.02×	180	1.10×	32	1.02×	84
FastV	1.04×	36	1.21×	291	1.38×	53	1.20×	122
SparseVLM	1.14×	4	**1.31**×	76	1.32×	9	**1.25**×	56
CDPruner	1.14×	19	1.08×	261	1.37×	26	1.11×	91
DivPrune	**1.18**×	20	1.25×	216	1.51×	24	**1.25**×	72
DART	1.17×	25	1.14×	346	1.35×	39	1.18×	103

Selection gains remain task specific, and Random can be competitive. At 95% correctness preservation, DivPrune leads on MMBench-EN, POPE, and GQA with speedups of $1.15\times$, $1.47\times$, and $1.18\times$, respectively, while SparseVLM leads on TextVQA at $1.23\times$. At 90% preservation, SparseVLM remains the fastest method on TextVQA ($1.31\times$) and ties DivPrune on GQA at $1.25\times$. In this block, Random achieves the highest reported speedup on POPE at $1.45\times$.

Stronger compression alone does not guarantee a speed benefit. At 95% preservation, VisionZip retains 289 of 750 tokens on TextVQA but reaches only $0.95\times$ speedup; on GQA, it retains 141 of 268 tokens and reaches only $0.97\times$. CDPruner is also slower than the dense model on TextVQA, with a speedup of $0.97\times$. On MMBench-EN, SparseVLM retains fewer tokens than DivPrune at both preservation targets, yet DivPrune is faster in both blocks. These results show why efficiency should be compared using measured inference time under the same correctness-preservation requirement, rather than inferred from the final number of retained tokens.

F Qualitative Analysis of Random-Pruning Sensitivity

F.1 Case Selection, Scoring, and Visual Encoding

Figures 5–7 show nine examples that the dense model answers correctly: one Easy, one Medium, and one Hard case for each of the three backbones. The LLaVA-1.5-7B examples come from VQAv2 val2014, while the LLaVA-NeXT-v1.6-Vicuna-7B and Qwen3-VL-8B examples come from GQA testdev-balanced. All images, masks, and predictions are taken from saved experimental records; we do not run any additional inference for this analysis. These cases are meant to illustrate the behavior we observe, rather than serve as a random sample for estimating the prevalence of each difficulty group or comparing backbones. Each case includes one successful and one unsuccessful random-pruning trial. As a result, examples with zero or ten successes do not appear in this paired display.

Scoring and traceability. Random-Pass@10 simply counts how many of ten random-pruning trials remain correct: 8–10 successes are labeled Easy, 4–7 Medium, and 0–3 Hard. For the VQAv2 cases shown here, we use the recorded consensus-score-positive rule; for GQA, we use normalized exact matching. All displayed successful answers have score 1, and all failures have score 0. Each figure row lists the sample, the retained and dense token counts, the reference answer, the dense-model prediction, and the trial numbers being shown. For VQAv2, we keep the original reference-answer frequencies rather than collapsing them into a single label. Table 13 reports the predictions from all ten trials for every case, including trials whose masks are not shown in the figures.

Table 13: **Complete random-trial answers for the nine displayed cases.** Columns index trials 1–10. Bold answers are correct under the saved scorer; plain answers are incorrect. Cases appear in the same order as the figure rows. Each dense prediction is correct.

Case	1	2	3	4	5	6	7	8	9	10
LLaVA-1.5-7B / VQAv2 / 32 tokens										
Easy	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	**1**	**1**
Medium	Food	**Ice cream**	**Ice cream**	**Ice cream**	**Ice cream**	Food	**Ice cream**	**Ice cream**	**Ice cream**	Coffee
Hard	White	White	White	White	**Orange**	Blue	Brown	White	Blue	White
LLaVA-NeXT-7B / GQA / 320 tokens										
Easy	**Sink**	**Sink**	Pipe	**Sink**	**Sink**	**Sink**	**Sink**	**Sink**	**Sink**	**Sink**
Medium	**Red**	White	White	Black	Yellow	Yellow	**Red**	White	**Red**	**Red**
Hard	Fence	Chair	**Table**	**Table**	**Table**	Chair	Chair	Fence	Chair	Chair
Qwen3-VL-8B / GQA / 20% retained										
Easy	bacon	**pancake**	**pancake**	**pancake**	**pancake**	**pancake**	**pancake**	**pancake**	**pancake**	**pancake**
Medium	**blue**	black	**blue**	**blue**	yellow	**blue**	**blue**	red	black	yellow
Hard	fruit	peaches	coffee	**tomatoes**	**tomatoes**	apples	peaches	coffee	coffee	coffee

Spatial encoding and its limits. Clear patches indicate retained token positions, while removed positions are covered by an 80% white overlay. These overlays show token selection after visual encoding, rather than pixel masking before the vision encoder. The LLaVA-1.5 visualizations keep the model's square padding. For LLaVA-NeXT, the global view and the high-resolution local view are shown separately; image-newline tokens count toward the token budget but do not correspond to image patches. The Qwen overlays use the saved spatial mapping, with the retained count rounded down from 20% of the native visual-token count. Matching views within the same case use the same scale, and the original image aspect ratio is preserved.

F.2 LLaVA-1.5-7B: Counting, Recognition, and Local Attributes

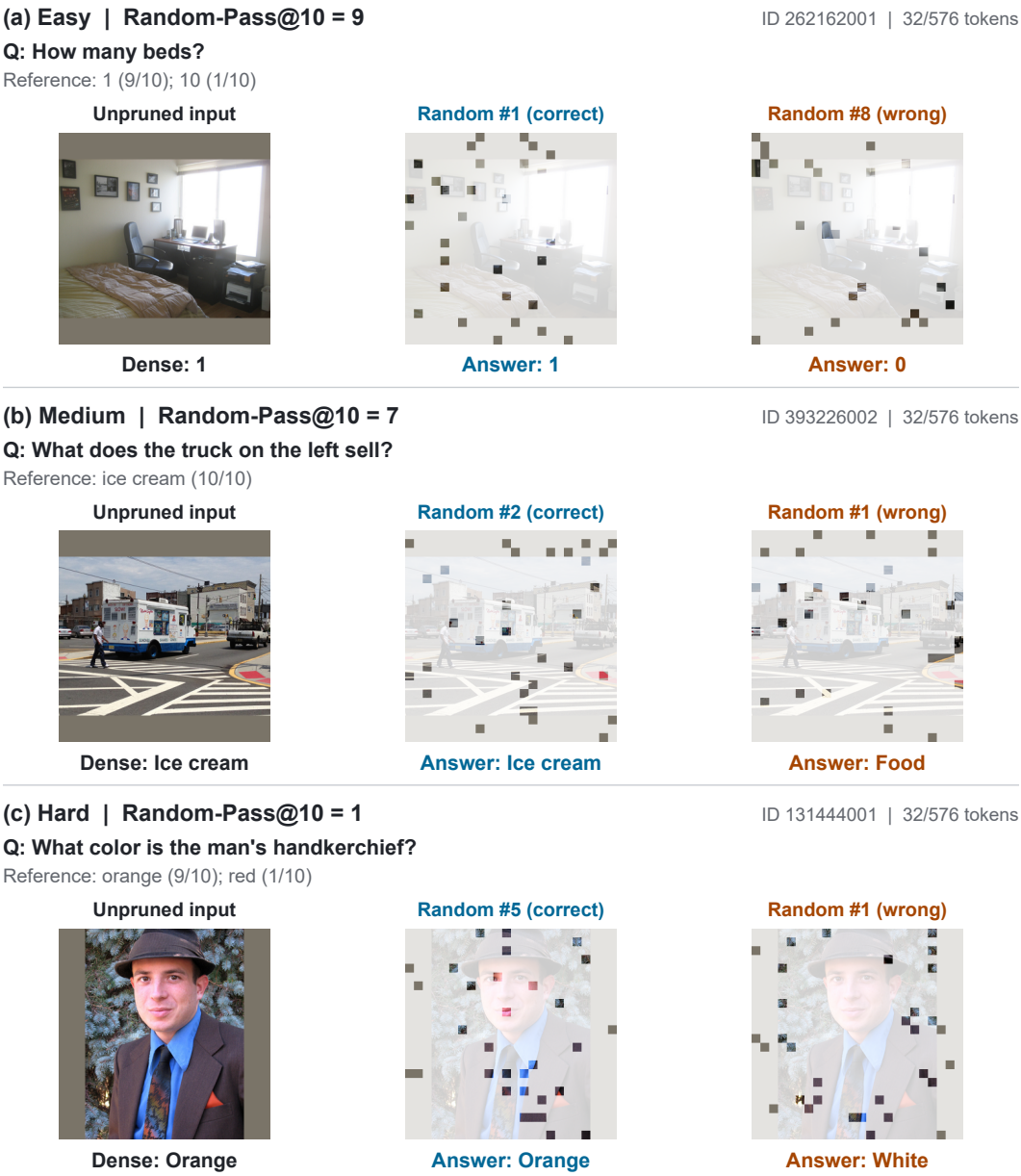


Figure 5: **Random pruning on LLaVA-1.5-7B at 32/576 tokens.** Each VQAv2 row pairs the unpruned model input with one successful and one unsuccessful saved mask at the same token budget. Blue and orange answer labels also indicate in text whether the prediction is correct. Clear patches are retained, while faded patches are removed. Trial numbers are one-based; all ten outcomes are reported in Table 13.

Reading the cases. The bed-counting question remains correct in nine of the ten trials, but one mask changes the answer from "1" to "0". The truck-recognition question succeeds in seven trials; when it fails, the model answers "Food" or "Coffee" instead of "Ice cream". The handkerchief-color question succeeds only once, and "White" is the most common incorrect answer. These examples show that the three questions differ substantially in how sensitive they are to random token selection.

F.3 LLaVA-NeXT-7B: Global and Local Token Views

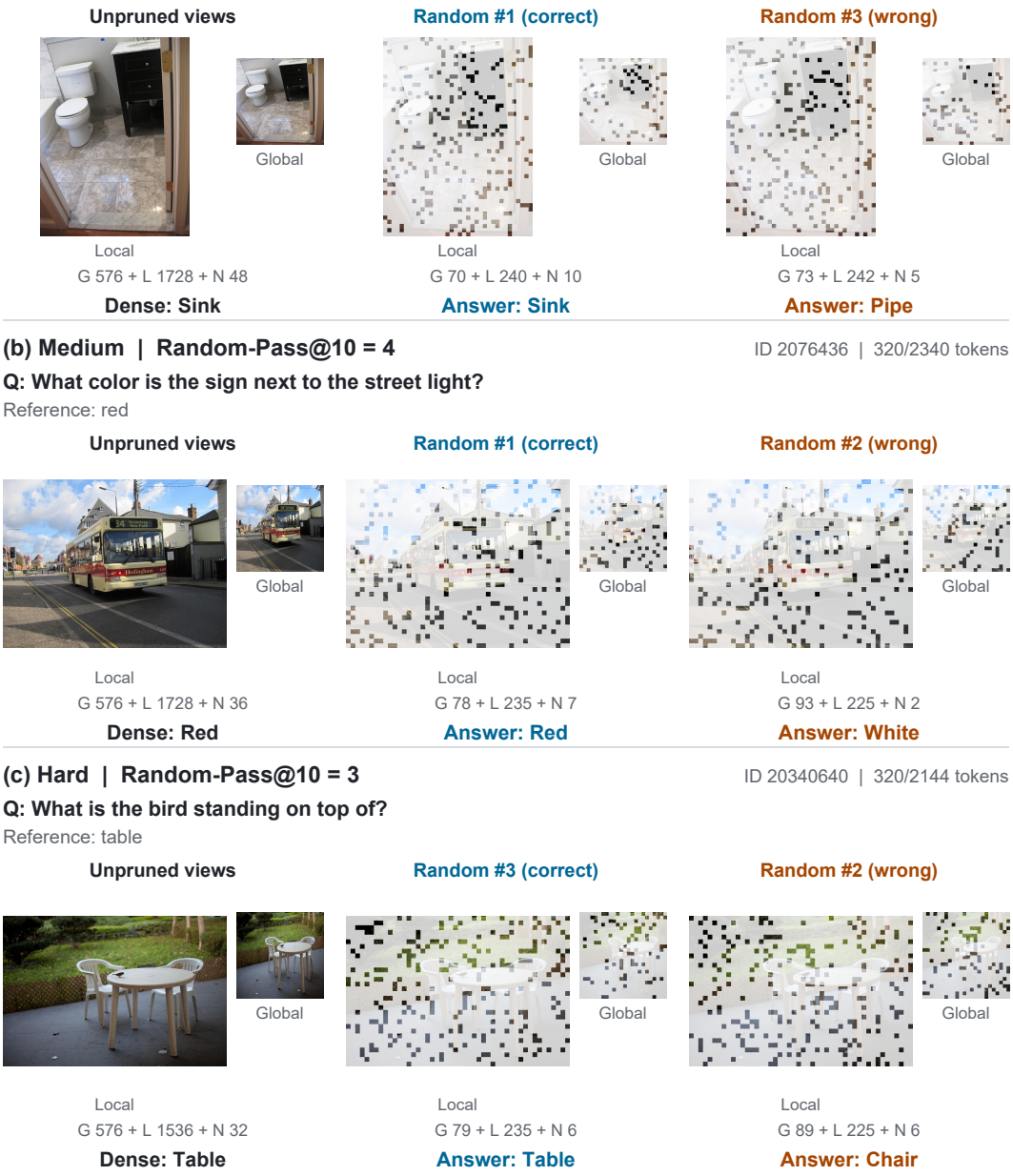


Figure 6: **Random pruning on LLaVA-NeXT-7B at 320 tokens.** The GQA examples show separate local and global views for the dense input and for both selected masks. G, L, and N denote the numbers of global, local, and image-newline tokens, respectively; together they sum to the corresponding token budget. Newline tokens do not correspond to spatial image patches. Clear positions are retained, while faded positions are removed.

F.4 QWEN3-VL-8B: RELATIVE BUDGETS AND ANSWER VARIABILITY

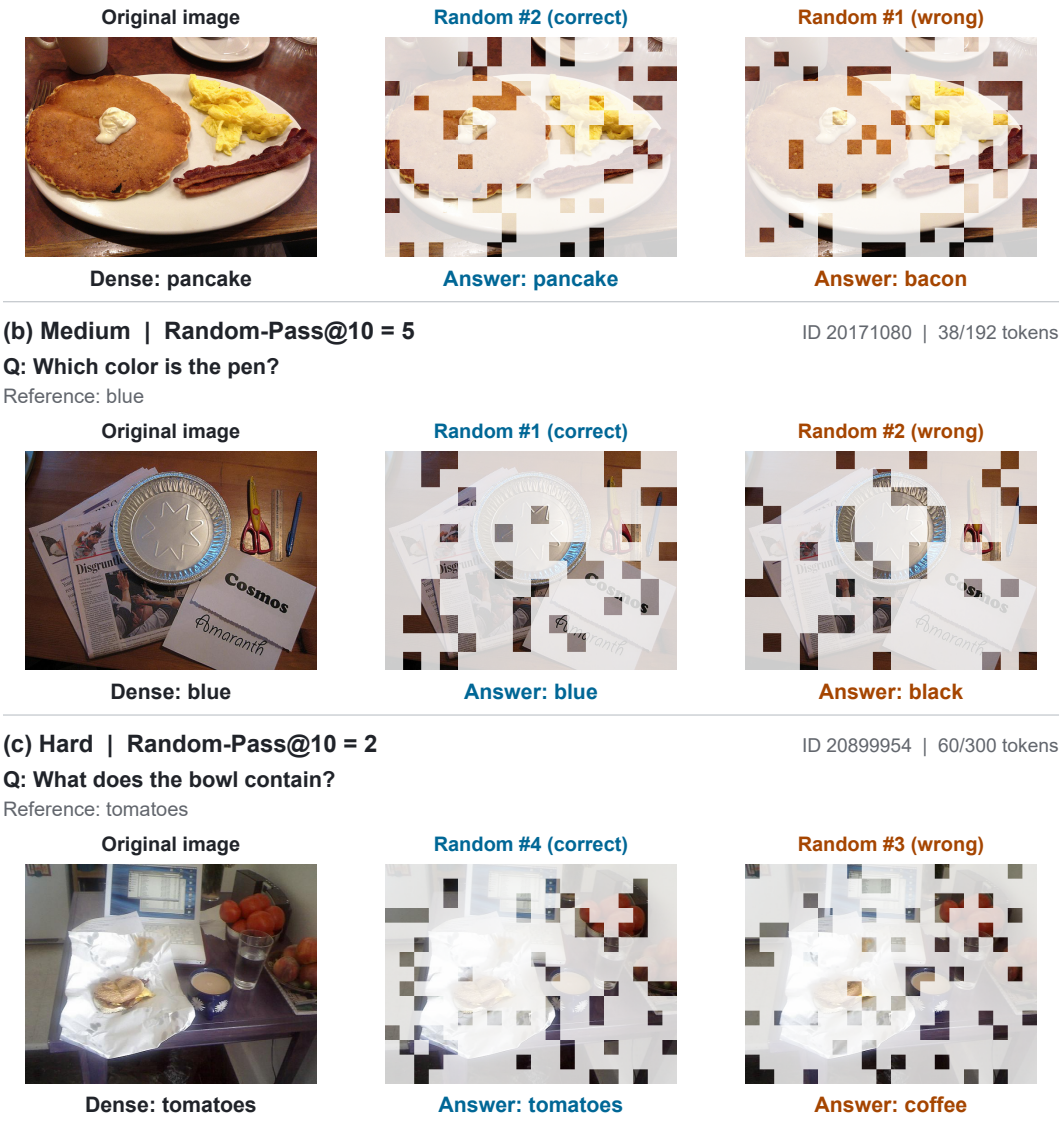


Figure 7: **Random pruning on Qwen3-VL-8B at a nominal 20% token budget.** The GQA examples pair the original image with saved overlays from one successful and one unsuccessful token-selection trial. The actual budgets are 60/300, 38/192, and 60/300 tokens; the middle case uses $\lfloor 0.2 \times 192 \rfloor = 38$. Clear positions are retained, while faded positions are removed. The original image is included only as a spatial reference and should not be read as a visualization of internal model features.

Reading the cases. The pancake question fails only once in ten trials, producing "bacon". The pen-color question produces "blue" in five trials, with the remaining answers split among black, yellow, and red. The bowl-content question produces "tomatoes" only twice, while "coffee" is the most common incorrect answer.

G Additional Analysis of Random-Pass Stratification

We examine how Random-Pass stratification varies across datasets and token budgets, and how its results compare when more random-pruning trials are used.

G.1 Difficulty Distributions across Token Budgets

Figures 8–10 show the difficulty composition of dense-correct examples for LLaVA-1.5 on eight datasets at 128, 64, and 32 retained visual tokens. At each budget, Random-Pass@10 assigns examples to Easy (8–10 correct trials), Medium (4–7), or Hard (0–3). Smaller budgets increase the Hard share on every dataset, but the shift is dataset-dependent: TextVQA's Hard share grows from 31.80% to 61.30%, whereas ScienceQA-IMG remains 82.58% Easy at 32 tokens. These differences motivate reporting difficulty-stratified results alongside aggregate performance.

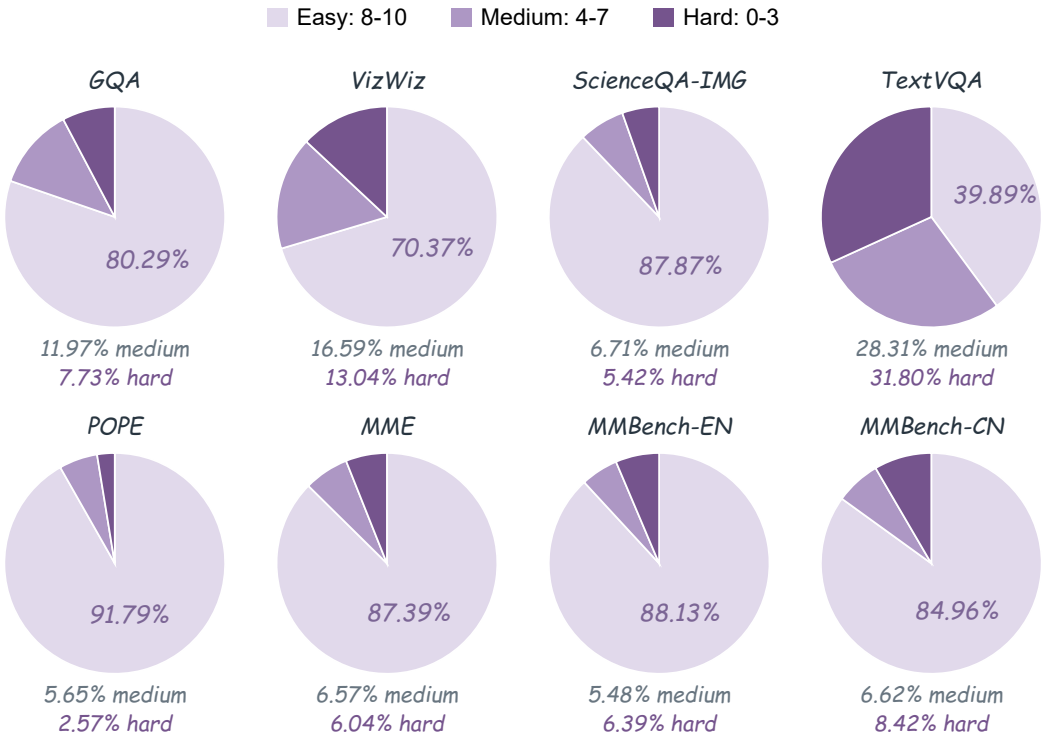


Figure 8: **Selection-difficulty distributions at 128 retained visual tokens.** Each pie reports the Easy, Medium, and Hard shares of dense-correct examples for LLaVA-1.5 on one dataset, using Random-Pass@10. The percentage inside each pie denotes Easy; the two values below denote Medium and Hard. Percentages may not sum to exactly 100% due to rounding.

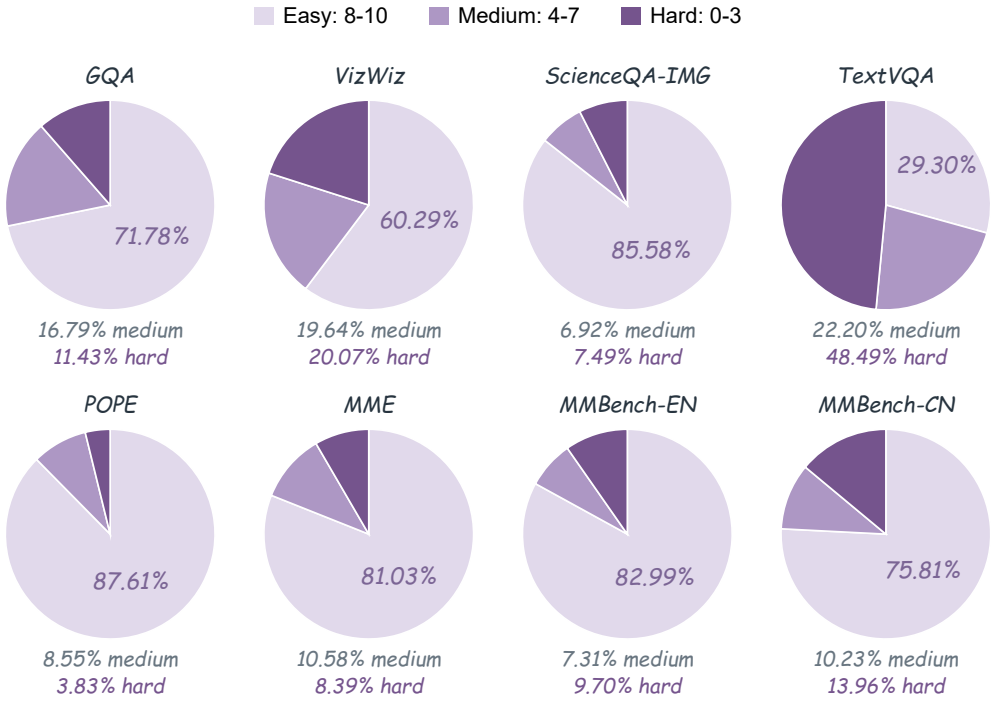


Figure 9: **Selection-difficulty distributions at 64 retained visual tokens.** Dataset order, colors, and annotations follow Figure 8. Relative to 128 tokens, every dataset has a smaller Easy share and a larger Hard share; TextVQA has the largest Hard share (48.49%).

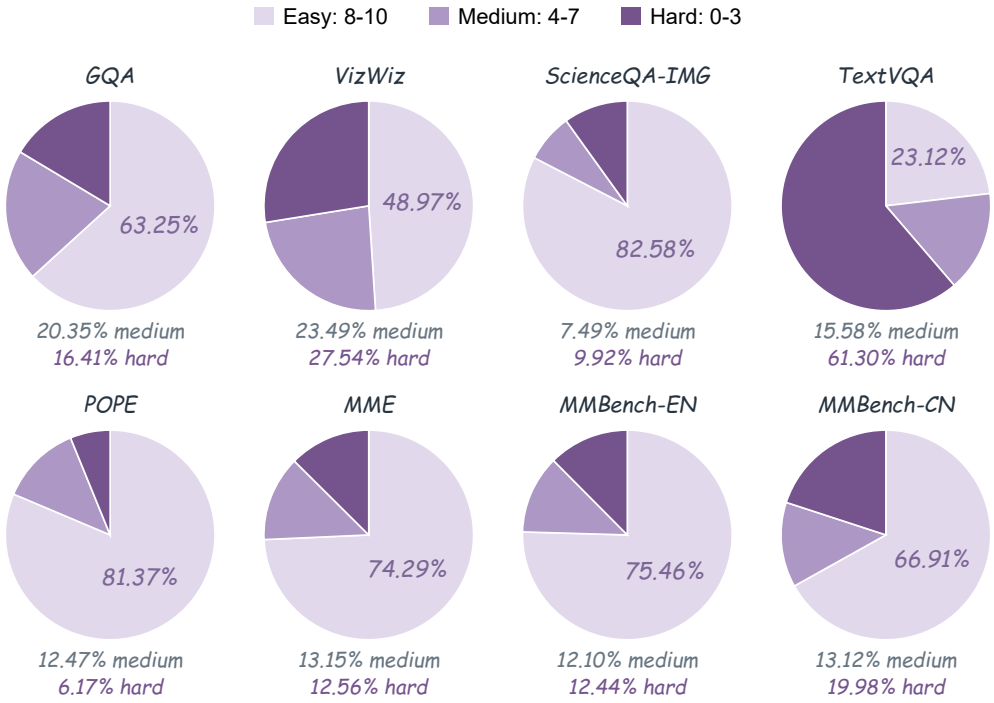


Figure 10: **Selection-difficulty distributions at 32 retained visual tokens.** Dataset order, colors, and annotations follow Figure 8. TextVQA is predominantly Hard (61.30%), while ScienceQA-IMG and POPE remain predominantly Easy (82.58% and 81.37%, respectively).

G.2 Sensitivity to the Number of Random Trials

Table 14 compares ten random-pruning trials with the cumulative set of 50 trials and with the independent 40 additional trials on GQA, TextVQA, and MMBench-EN at the same three budgets. The independent comparison avoids sharing trials with the ten-trial reference.

Table 14: **Random-Pass overlap across trial counts (%).** For each retained-token budget, "10 vs. 50" compares the original ten trials with all 50 trials; "10 vs. 40" compares them with the independent additional 40 trials. The overall values are unweighted means across the nine dataset–budget conditions.

	128 tokens		64 tokens		32 tokens	
Dataset	10 vs. 50	**10 vs. 40**	10 vs. 50	**10 vs. 40**	10 vs. 50	**10 vs. 40**
GQA	91.0	90.2	90.0	87.4	89.4	87.2
TextVQA	84.4	81.2	87.4	84.6	90.8	87.8
MMBench-EN	97.2	96.8	95.2	94.2	93.6	92.4
Overall: 10 vs. 50 = 91.0%; **10 vs. 40 = 89.1%**						

The mean overlap is 91.0% with the cumulative 50 trials and 89.1% with the independent 40 trials. The independent overlap ranges from 81.2% to 96.8%, indicating that the observed stability varies across datasets and budgets.