PRUNING BY CONSEQUENCE: INTERVENTION-GUIDED VISUAL TOKEN SELECTION FOR VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free visual token selection reduces the inference cost of vision–language models by keeping a small subset of image tokens. Existing selectors rank tokens by attention, similarity, saliency, or diversity—proxies for the quantity that matters at deployment, namely whether the compressed model still produces the answer the uncompressed model would. We propose CIG (Cluster Intervention Gain), which scores a candidate subset by that quantity directly: it maximises $J(S) = -\mathrm{KL}(p(y \mid V, x) \,\|\, p(y \mid V_S, x))$ with real subset prefills and a cluster-level greedy search, and requires neither labels nor training. Under a controlled *exact-576* protocol on InternVL3-8B, Qwen2.5-VL-7B, InternVL3.5-4B and Qwen3-VL-4B, CIG outperforms every published architecture-portable selector in all 24 backbone–budget settings, with paired intervals excluding zero for every backbone at $k \leq 64$; the ordering holds on a held-out slice disjoint from all configuration data and under free-form generation scoring on GQA and TextVQA. Its margin over the best heuristic grows from $+0.99$ pp at $k = 192$ to $+11.57$ pp at $k = 8$, where it retains at least 93.3% of full-token accuracy at $72\times$ compression. A batched variant matches the sequential selector within 0.65 pp while selecting 1.6–$4.7\times$ faster, and under serving-scale batching the shorter KV cache reduces end-to-end latency by up to 17.9% on a memory-pressured 8B workload, selection cost included. Ablations attribute most of the gain to the intervention objective rather than to the clustering heuristics that make the search tractable.

1 INTRODUCTION

Vision–language models spend a large fraction of inference on image tokens: LLaVA-1.5-7B represents a 336×336 image with 576 visual tokens (Liu et al., 2024a), and high-resolution variants produce several thousand (Liu et al., 2024b). Reducing that prefix lowers prefill work and, more importantly for long or batched decoding, the size of the KV cache. That much of the prefix is redundant is well documented; the harder question is which k tokens can be kept without changing the model's prediction.

Most training-free selectors answer this question indirectly, through decoder attention (FastV, Chen et al., 2024), vision-encoder `[CLS]` attention (FasterVLM, VisionZip, Zhang et al., 2024b; Yang et al., 2025), diversity (DivPrune, CDPruner, Alvar et al., 2025; Zhang et al., 2025b) or multimodal coverage (MMTok, Dong et al., 2026). These signals are inexpensive, but they are proxies for the quantity that ultimately matters, whether the compressed model behaves like the uncompressed one, and their rankings are sensitive to architecture, position and task, with random pruning a surprisingly strong control (Peng et al., 2025; Wen et al., 2025a).

We evaluate the consequence of removing tokens directly. For a retained subset S, we define predictive faithfulness as

$$J(S) \;=\; -\,\mathrm{KL}\big(p(y \mid V, x) \;\big\|\; p(y \mid V_S, x)\big)\,, \tag{1}$$

where the full model supplies the reference distribution. The objective is label-free and depends on no attention map, encoder special token or pruning schedule. Computing it for every subset is infeasible, so CIG groups visual tokens into $C = 8$ density-peak clusters, evaluates candidate cluster insertions

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.

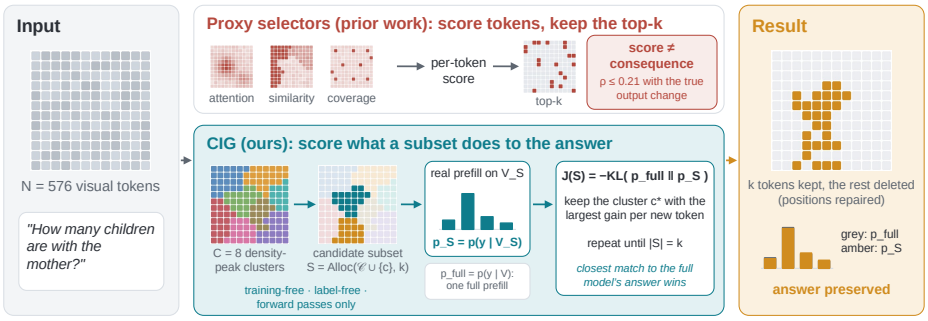


Figure 1: **Pruning by consequence.** Proxy selectors rank tokens by a score that stands in for importance; CIG instead asks what a candidate subset does to the model's answer: it runs real prefills on cluster-induced subsets, scores each by $-\mathrm{KL}(p(y \mid V, x) \,\|\, p(y \mid V_S, x))$, and greedily keeps the clusters with the largest gain per token until k tokens remain.

with real prefills, and realises the requested budget with a round-robin allocator: the search runs over a small candidate set while the score is measured on the deployed model.

Our primary evaluation uses a single *exact-576* protocol so that a budget has the same meaning across InternVL3-8B, Qwen2.5-VL-7B, InternVL3.5-4B and Qwen3-VL-4B. Six budgets on four datasets yield 24 backbone–budget comparisons, in every one of which CIG is best among selectors that transfer without architecture-specific signals; the margin over the strongest heuristic grows from $+0.99$ pp at $k = 192$ to $+11.57$ pp at $k = 8$, where CIG still retains at least 93.3% of full-token accuracy, and paired intervals exclude zero for every backbone at $k \leq 64$. We additionally compare with published LLaVA-1.5 selectors, evaluate a batched search, measure serving-scale latency, and isolate the contribution of the objective through component ablations and equal-information controls.

Contributions. **(i)** We formulate training-free visual token selection as direct preservation of the uncompressed model's predictive distribution and optimise that criterion with cluster-level intervention search. **(ii)** We provide a controlled four-backbone, six-budget exact-576 evaluation with true token deletion and paired statistics, and confirm the ordering on a held-out slice, under free-form generation scoring, and against equal-information controls that see the same reference prefill. **(iii)** We study the practical trade-off between selection quality and cost: batched probing closely matches the sequential selector, serving-scale compression can improve latency under memory pressure, and ablations identify the intervention objective as the main source of the accuracy gain.

2 Related Work

Visual token reduction. Training-free methods reduce visual tokens either inside the language model, where FastV, SparseVLM, PyramidDrop and VTW progressively remove tokens across layers (Chen et al., 2024; Zhang et al., 2025c; Xing et al., 2025; Lin et al., 2025), or before it, using vision-encoder saliency or merging (Yang et al., 2025; Zhang et al., 2024b; 2025a), similarity-based merging (Bolya et al., 2023; Shang et al., 2025), hierarchical attention (Arif et al., 2025; Liu et al., 2025), fitted pruning schedules (Ye et al., 2025), or coverage and diversity objectives (Dong et al., 2026; Alvar et al., 2025; Zhang et al., 2025b). CIG is a pre-LLM selector and therefore emits an explicit prefix of at most k tokens, making token budget and KV-cache length directly measurable across compatible backbones.

From proxy scores to output preservation. Several recent methods move beyond a single attention or similarity score: CoIn combines informativeness and coverage (Du et al., 2026), ApET uses linear approximation error (Ma et al., 2026), ConsensusDrop fuses visual and cross-modal saliency (Parikh et al., 2026), MetaCompress learns a prompt-agnostic compression for multi-turn use (Wang et al., 2026), ZOO-Prune estimates token sensitivity with zeroth-order perturbations (Kim et al., 2026), and Fourier-VLM compresses features in the frequency domain without selecting a subset (Wang et al., 2025a); see Shao et al. (2026) for a taxonomy. CIG differs by evaluating the deployed model on

candidate subsets and scoring the resulting output distribution directly, in the spirit of ablation-based faithfulness tests in interpretability (Hooker et al., 2019; Jain & Wallace, 2019); the duplication analysis of DART explains why distinct retained subsets can preserve similar outputs (Wen et al., 2025b).

3 METHOD

3.1 PREDICTIVE FAITHFULNESS AS THE SELECTION OBJECTIVE

Let an image produce projected visual tokens $V = (v_1, \ldots, v_N)$ in the language model's embedding space and let x be the tokenised prompt; for $S \subseteq [N]$ write V_S for the subsequence in raster order. Given an answer support $\mathcal{A} = (a_1, \ldots, a_m)$, let $T(a_j)$ collect the tokeniser variants of option a_j (for multiple choice `"A"` and `" A"`; for yes/no also capitalised forms) and $\ell(V_S, x)$ the next-token logits at the first answer position. The option distribution is

$$p_S(j) \;=\; \frac{\exp\big(\mathrm{LSE}_{t\in T(a_j)}\,\ell_t(V_S,x)\big)}{\sum_{j'}\exp\big(\mathrm{LSE}_{t\in T(a_{j'})}\,\ell_t(V_S,x)\big)}, \qquad \mathrm{LSE}_{t\in T}\,\ell_t = \log\sum_{t\in T} e^{\ell_t}, \tag{2}$$

where the log-sum-exp merges surface variants so tokenisation does not leak into the objective. Writing $p_\star = p_{S=[N]}$ (all visual tokens) and $p_\varnothing = p_{S=\emptyset}$ (none), our objective is

$$J_1(S) \;=\; -\,\mathrm{KL}(p_\star \,\|\, p_S)\,, \tag{3}$$

maximised at 0 when the retained subset reproduces the uncompressed model's answer distribution exactly. The objective is label-free, and the direction of the divergence matters: the reverse KL would tolerate a confidently wrong subset that concentrates mass where $p_\star$ is thin. Section 6 compares J_1 with a confidence gain $J_2(S) = H(p_\varnothing) - H(p_S)$, a mixed form $J_3(S) = \mathrm{KL}(p_S\|p_\varnothing) - \lambda\,\mathrm{KL}(p_\star\|p_S)$, and a label-using upper bound $J_{\mathrm{oracle}}(S) = p_S(y^\star)$ used only as a headroom marker; J_1 was fixed on a screening split before the main experiments (Appendix P).

For open-ended tasks (GQA, TextVQA and captions), where no fixed option list exists, we construct a small answer support from the highest-scoring surface forms under the blind and full visual conditions. Thus J_1 is a first-answer-token faithfulness objective for open-ended generation rather than a KL over complete sequences; Section 5.1 tests it against free-form generation scoring.

3.2 CLUSTER-LEVEL GREEDY INTERVENTION SEARCH

Optimising Equation 3 over $\binom{N}{k}$ subsets with one forward pass per evaluation is hopeless, and token-level greedy insertion would still need $O(Nk)$ prefills. We reduce the search space in three steps.

(1) Clustering. Tokens are grouped into C clusters by density-peaks clustering with kNN density (Rodriguez & Laio, 2014; Du et al., 2016) on the ℓ_2-normalised projected tokens $\bar{v}_i$. Density is $\rho_i = \exp(-\frac{1}{n}\sum_{j\in\mathcal{N}(i)}\|\bar{v}_i - \bar{v}_j\|^2)$ over the $n=5$ nearest neighbours, $\delta_i = \min_{j:\rho_j>\rho_i}\|\bar{v}_i - \bar{v}_j\|^2$, centres are the top-$C$ of $\rho_i\delta_i$, and remaining tokens attach to the nearest labelled point of higher density, with ties broken by a fixed seed. Appending a weighted grid coordinate γg_i to $\bar{v}_i$ changes the four-task macro by at most 0.35 pp (Table 7), so $\gamma = 0$ throughout.

(2) Budget realisation. Given an ordered set of selected clusters $\mathcal{C}$ we emit $\textsc{Alloc}(\mathcal{C}, k)$: members of each selected cluster are queued in descending order of a member prior and the queues drained round-robin by increasing depth. ALLOC draws only from selected clusters, so it returns $\min(k, \sum_{c\in\mathcal{C}}|c|)$ tokens and underfills whenever they are collectively smaller than k, in which case the search continues; and because the queues are re-drained at every step, admitting a cluster can change which members earlier clusters contribute. The member prior is the maximum cosine similarity between the projected token and any prompt-token embedding, the cluster prior its within-cluster mean; priors order candidates and members and thereby define the candidate space, but never score subsets.

Algorithm 1: CIG: cluster-level intervention-gain token selection (training-free, label-free). ALLOC may underfill, which is what keeps the outer loop running; Appendix A explains the termination behaviour and gives a worked trace.

Input: projected tokens V (N), prompt x, budget k, clusters C, fan-out M
Output: token index set S with $|S| = \min(k, N)$

Function `Alloc` $(\mathcal{C}, k)$ **:**
  queue the members of each $c \in \mathcal{C}$ by decreasing π^{mem}; drain the queues round-robin by increasing depth until $|S| = k$ or all queues are exhausted;
  return S ; // $|S| = \min\big(k,\ \sum_{c\in\mathcal{C}} |c|\big)$ --- **may underfill**
$p_\star \leftarrow \textsc{Read}(V, x)$; $p_\varnothing \leftarrow \textsc{Read}(\emptyset, x)$; // 2 reference prefills
$\{\mathcal{K}_1..\mathcal{K}_C\} \leftarrow$ DPC-KNN(V, C); $\pi^{\mathrm{mem}}, \pi^{\mathrm{clu}} \leftarrow \textsc{CrossModalPrior}(V, x)$;
$\mathcal{C} \leftarrow \emptyset$; $S \leftarrow \emptyset$; $p_S \leftarrow p_\varnothing$;
while $|S| < k$ ***and*** $\mathcal{C} \neq \{1..C\}$ **do**
  $\mathcal{B} \leftarrow$ top-M clusters of π^{clu} not in $\mathcal{C}$;
  foreach $c \in \mathcal{B}$ **do**
    $S_c \leftarrow$ `Alloc` $(\mathcal{C} \cup \{c\}, k)$; // re-derived from scratch
    $p_{S_c} \leftarrow \textsc{Read}(V_{S_c}, x)$; // real subset prefill
    $\Delta_c \leftarrow \big(J_1(S_c) - J_1(S)\big) / \max(1, |S_c \setminus S|)$;
  end
  $c^\star \leftarrow \arg\max_{c\in\mathcal{B}} \Delta_c$; $\mathcal{C} \leftarrow \mathcal{C} \cup \{c^\star\}$; $S \leftarrow S_{c^\star}$; $p_S \leftarrow p_{S_{c^\star}}$;
end
$S \leftarrow$ `Alloc` $(\mathcal{C}, k)$; **if** S *changed* **then** $p_S \leftarrow \textsc{Read}(V_S, x)$;
return S;

(3) Greedy insertion with gain density. Starting from $\mathcal{C}_0 = \emptyset$, $S_0 = \emptyset$, $p_{S_0} = p_\varnothing$, step t considers the M highest-prior unselected clusters, forms $S_c = \textsc{Alloc}(\mathcal{C}_{t-1} \cup \{c\}, k)$ for each, reads p_{S_c} by a real prefill, and selects the candidate maximising *gain density*

$$\Delta_c \;=\; \frac{J_1(S_c) - J_1(S_{t-1})}{\max\big(1,\ |S_c \setminus S_{t-1}|\big)}. \tag{4}$$

Normalising by newly admitted tokens prevents a large cluster from winning merely by consuming more budget, in the spirit of cost-benefit greedy maximisation (Nemhauser et al., 1978); J_1 is not submodular in general, so the rule is a heuristic whose value Section 6 establishes empirically. Algorithm 1 gives the complete procedure.

Search depth. The loop stops once the selected clusters can realise budget k. With $C = 8$ and $N = 576$ a cluster holds 72 tokens on average, so low-budget runs often end after the first candidate round, whereas larger budgets or a small leading cluster trigger further insertions (Appendix A traces both regimes); the search is shallow by design, with the number of rounds set by the budget rather than by a convergence criterion.

Cost. CIG requires a full-reference prefill, a blind prefill and several candidate prefills; at $k = 32$ the measured mean is 401 candidate visual tokens per request, and the largest single term is the 576-token reference prefill itself, the price of knowing $p_\star$ (Appendix O). Selection therefore adds computation before any decoding saving is realised, so the method is most attractive when the shorter KV cache matters, as in long or batched generation; Section 5 evaluates both the sequential selector and a batched approximation.

4 EXPERIMENTAL SETUP

Primary protocol: exact-576, four backbones. We fix native visual tokens to $N = 576$ on every backbone, so a budget k has identical meaning across architectures and is comparable with the 576-token setting the literature uses on LLaVA-1.5; input resolution is set per model to realise that count (672 px for all backbones except Qwen3-VL-4B at 768 px). Budgets are $k \in \{8, 16, 32, 64, 128, 192\}$; datasets are POPE, MMBench-EN-dev, SEED-Bench-image and ScienceQA-IMG at the first 500 examples each (Li et al., 2023; Liu et al., 2024c; Li et al., 2024; Lu et al., 2022); dtype BF16; selection

Table 1: **Main result: exact-576 native protocol.** Four-task macro accuracy (%) with tokens deleted rather than masked. *Best heur.* is the strongest of Random/MMTok/DivPrune/CDPruner *in that cell*; the ± after Δ is the larger distance from Δ to either end of that cell's *paired* bootstrap 95% interval (10,000 draws, best opponent re-selected inside every resample; intervals in Appendix C), and Δ is **bold** where that interval excludes zero. Per-method and per-dataset rows: Appendix B; FastV, VisionZip, SparseVLM and PyramidDrop are not architecture-comparable here (Appendix F).

Backbone	Method	k=8	k=16	k=32	k=64	k=128	k=192	Full
InternVL3-8B	Best heur.	70.80	74.45	79.45	83.05	86.05	86.55	
	CIG	**81.80**	**83.55**	**85.40**	**86.25**	**86.40**	**87.30**	87.65
	Δ	**+11.00±2.50**	**+9.10±2.50**	**+5.95±2.35**	**+3.20±2.40**	+0.35±2.15	+0.75±2.15	
Qwen2.5-VL-7B	Best heur.	68.75	72.90	76.75	81.30	83.05	84.20	
	CIG	**80.55**	**82.35**	**84.20**	**84.90**	**85.30**	**85.75**	85.90
	Δ	**+11.80±2.60**	**+9.45±2.55**	**+7.45±2.55**	**+3.60±2.30**	+2.25±2.45	+1.55±2.45	
InternVL3.5-4B	Best heur.	65.75	71.70	73.90	77.25	80.00	82.20	
	CIG	**78.65**	**80.05**	**80.75**	**81.50**	**82.50**	**82.45**	84.20
	Δ	**+12.90±2.80**	**+8.35±2.55**	**+6.85±2.85**	**+4.25±2.85**	+2.50±2.80	+0.25±2.45	
Qwen3-VL-4B	Best heur.	71.65	76.30	80.40	83.60	85.40	86.00	
	CIG	**82.25**	**84.15**	**86.05**	**86.55**	**86.90**	**87.40**	87.95
	Δ	**+10.60±2.60**	**+7.85±2.45**	**+5.65±2.25**	**+2.95±2.15**	+1.50±2.10	+1.40±2.20	
Mean Δ over four backbones		**+11.57**	**+8.69**	**+6.48**	**+3.50**	**+1.65**	**+0.99**	
Cells whose paired interval excludes zero		4/4	4/4	4/4	4/4	0/4	0/4	
CIG retention of Full (min.)		93.3%	95.1%	95.9%	96.8%	98.0%	97.9%	100%

*Against each **fixed** baseline over all 24 cells, paired cluster bootstrap over (backbone, task);* CIG *wins* $24/24$ *against every one:*
vs Random $+7.39$ $[+6.42, +8.36]$; vs MMTok $+5.69$ $[+4.78, +6.61]$;
vs DivPrune $+7.34$ $[+6.19, +8.57]$; vs CDPruner $+8.81$ $[+7.54, +10.11]$.

deterministic given seed 42. Tokens are removed rather than masked, which requires positional repair—contiguous re-indexing for InternVL, synchronous pruning of the Qwen multimodal rotary grid (Su et al., 2024; Wang et al., 2024). Backbones are InternVL3-8B and InternVL3.5-4B (Zhu et al., 2025; Wang et al., 2025b), and Qwen2.5-VL-7B and Qwen3-VL-4B (Bai et al., 2025b;a). The grid contains $4 \times 6 \times 4 \times 500 = 48{,}000$ sample–budget evaluations. CIG uses the same configuration throughout ($C = 8$, J_1, $M = 8$); only the LLaVA-NeXT transfer study re-selects C (Appendix L).

Secondary protocol and baselines. To compare against the largest possible set of *published* selectors we also run the customary setting: LLaVA-1.5-7B (Liu et al., 2023; 2024a) with CLIP ViT-L/14-336 (Radford et al., 2021), BF16, batch 1, seed 42, greedy decoding, `lmms-eval` (Zhang et al., 2024a), and eight complete splits totalling $50{,}431$ items (Hudson & Manning, 2019; Yue et al., 2024; Singh et al., 2019; Fu et al., 2023), itemised in Appendix H; MME is a native raw total, excluded from every percentage macro. Results from the two protocols are not pooled. All hyper-parameters were fixed on a SCREEN split, the first 500 items of each task on LLaVA-1.5, which is contained in FULL and coincides with the items of the exact-576 grid; Section 5.1 therefore repeats the grid on a disjoint held-out slice (items 501–1,000). We integrate six published selectors using their released implementations, plus MMTok and a random control, and report which are admissible under each protocol: under exact-576 only Random, MMTok, DivPrune and CDPruner transfer directly, while FastV, VisionZip, SparseVLM and PyramidDrop each need a signal or schedule that the target architectures do not expose and are marked NA (Appendix F gives the reason for each). All six appear in the LLaVA comparisons, where three cannot structurally reach low budgets (Appendix I).

Statistics. Every comparison is paired per sample, with $10{,}000$-draw stratified paired bootstrap 95% intervals (Efron & Tibshirani, 1994) and exact McNemar tests (McNemar, 1947), MMBench circular variants collapsed first; the exact-576 grid has $2{,}000$ pairs per cell and a cluster bootstrap over $(\text{backbone}, \text{task})$ for aggregates (Appendix C).

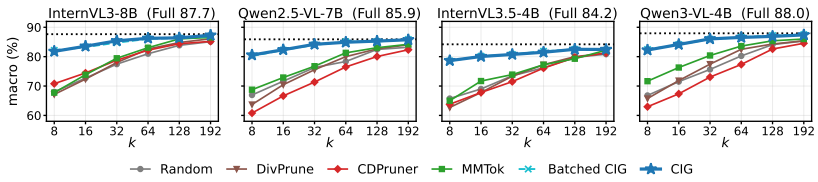


Figure 2: **Budget curves under exact-576**, four-task macro against the retained budget k. Heuristics degrade steeply below $k = 64$ and reorder between backbones; CIG stays near the full-token ceiling (batched variant indistinguishable at this scale).

Table 2: **Free-form generation on the held-out slice** (items 501–1,000, exact-576, $k = 32$, $n = 500$ per cell; greedy decoding scored with the `lmms-eval` GQA and TextVQA metrics). Best pruned method in bold; †: paired interval of CIG against MMTok excludes zero. Other budgets, SCREEN slice, fidelity: Appendix E.

	GQA (exact match, %)						TextVQA (VQA accuracy, %)					
Backbone	Full	Random	MMTok	DivPrune	CDPruner	CIG	Full	Random	MMTok	DivPrune	CDPruner	CIG
InternVL3-8B	63.8	49.6	56.2	54.8	53.4	**58.4**	70.9	25.2	39.6	41.3	25.4	**45.8**†
Qwen2.5-VL-7B	66.0	52.0	58.0	54.8	48.8	**62.0**†	83.3	35.1	51.0	45.9	31.3	**61.6**†
InternVL3.5-4B	59.4	49.4	51.6	51.6	47.0	**56.2**†	65.0	21.4	32.8	34.9	26.5	**36.8**†
Qwen3-VL-4B	65.6	55.8	58.4	56.0	48.8	**63.8**†	78.5	30.4	49.9	47.2	34.1	**53.0**

5 RESULTS

5.1 FOUR BACKBONES UNDER ONE EXACT-576 PROTOCOL

Across the 24 backbone–budget cells, CIG has the highest macro accuracy among the five architecture-portable selectors. Relative to the strongest heuristic selected within each cell, paired intervals exclude zero for all four backbones at $k \leq 64$; at $k \geq 128$ the point estimates remain positive but the intervals overlap zero, and against each fixed baseline the cluster-bootstrap mean difference is positive in all 24 cells ($+5.69$ to $+8.81$ pp; Table 1). The advantage is largest under aggressive compression: at $k = 8$ ($72\times$ compression) CIG retains 93.3–93.8% of full-token macro accuracy against 78.1–81.5% for the strongest heuristic, and the gap narrows toward parity as the budget grows. The identity of that strongest heuristic is itself unstable across backbones and budgets (Figure 2; Appendix B), so committing to any single proxy risks inheriting the wrong one for a given deployment, whereas the intervention objective is defined by the deployed model itself.

Equal-information controls. The heuristics above see only input-level features, whereas CIG sees the output of a reference prefill. A gradient-saliency selector given the same prefill (GRADFULL: input$\times$gradient norm of the full model's own argmax log-probability, label-free) beats the best input-level heuristic in all 24 cells ($+0.45$ to $+7.45$ pp), so output-level information is what the heuristics lack; CIG in turn beats GRADFULL in all 16 cells at $k \leq 64$ ($+0.1$ to $+11.0$ pp) and is within 1.2 pp of it at $k \geq 128$. Scoring clusters by what their deletion from the full context costs (KNOCKOUT) trails CIG at every budget ($+0.3$ to $+12.6$ pp; Appendix G).

Held-out slice and free-form generation. Repeating the grid on items 501–1,000 of each task, disjoint from every item used for configuration, reproduces the margins: pooled over $k \in \{32, 64\}$, CIG leads the best heuristic by $+4.05$ pp on InternVL3-8B and $+4.23$ pp on Qwen2.5-VL-7B (SCREEN: $+4.58$, $+5.53$) and leads every heuristic in all eight task cells (Appendix D). Nor does the result depend on the first-token readout: scoring free-form greedy generations with the `lmms-eval` GQA and TextVQA metrics on the same slice (Table 2) gives the same ordering at $k = 32$, with CIG best in all eight backbone–task cells, its paired interval against MMTok excluding zero in six, and its generations reproducing the full model's own answer most often (78–87% on GQA against 65–77% for MMTok). Readout and generation scoring agree on $\geq 95\%$ of items in 90 of 96 multiple-choice conditions (Appendix M); at larger budgets MMTok overtakes CIG on TextVQA (Appendix E), the OCR boundary of Section 5.4.

Table 3: **Batched probing preserves accuracy while selecting 1.6–$4.7\times$ faster.** Jaccard and response match are against the exact sequential selector; Δacc is batched minus exact four-task macro; the fidelity ranges span $k \in \{8, \ldots, 192\}$ (per-budget rows in Appendix B); speed-up (sequential $\div$ batched selection time) spans budgets and both settings (single request and serving-scale, Appendix N).

Backbone	Token Jaccard	Response match	Δacc (pp)	Fidelity gate (pass / 6)	Speed-up
InternVL3-8B	0.664 – 0.762	0.970 – 0.987	-0.65 to $+0.35$	0 / 6	1.6–2.2×
Qwen2.5-VL-7B	0.803 – 0.887	0.989 – 0.994	-0.35 to $+0.20$	6 / 6	1.8–2.2×
InternVL3.5-4B	0.822 – 0.914	0.985 – 0.992	-0.25 to $+0.35$	6 / 6	3.2–3.7×
Qwen3-VL-4B	0.845 – 0.923	0.993 – 0.995	-0.10 to $+0.30$	6 / 6	3.4–4.7×

Table 4: **LLaVA-1.5-7B, eight full splits,** $k = 32$ (5.6% of 576 tokens), a standard setting in the visual-token pruning literature. Accuracy (%) except MME (native raw total, excluded from the macro). Paired bootstrap 95% intervals against CIG in the last column; † marks a paired difference whose interval excludes zero. FastV, SparseVLM and PyramidDrop cannot reach $k = 32$ under their published schedules (Appendix I).

Method	POPE	MMB	SEED	SQA	GQA	MMMU	TextVQA	MME	Avg$_7$	95% CI on Δ
Full (576)	86.86	64.18	66.16	69.61	61.88	35.44	45.82	1854.9	61.42	—
Random	73.61	46.05	53.30	66.58	52.31	33.78	17.20	1424.2	48.98	—
VisionZip	75.03	57.39	53.22	68.77	51.88	**35.11**	37.37	1576.1	54.11	—
DivPrune	84.02	57.13	57.73	69.26	55.58	34.56	35.18	1610.2	56.21	—
MMTok	84.99	59.02	59.78	68.96	56.47	34.89	**37.41**	1678.0	57.36	$[+0.38, +1.33]^\dagger$
CDPruner	**87.52**	**60.74**	60.91	68.62	57.35	35.00	36.26	1674.9	58.06	$[-0.31, +0.64]$
CIG (ours)	85.58	**60.74**	**64.43**	**69.66**	**59.24**	35.00	32.91	**1795.9**	**58.22**	—

5.2 Batched intervention search

Batching candidate prefills reduces selection time while preserving accuracy (Table 3): across all 24 cells the batched variant stays within 0.65 pp of the sequential selector and is 1.6–$4.7\times$ faster. Accuracy parity does not imply path identity, however. We gate each configuration on token-set Jaccard ≥ 0.8 and response match ≥ 0.98 against the sequential selector: three backbones pass at every budget, whereas InternVL3-8B fails at all six, its Jaccard falling to 0.664 while responses still agree on 97.0–98.7% of items. The mechanism is numerical: BF16 reductions differ with batch shape, perturb near-ties in Equation 4 and send the greedy search down a different path, so the batched implementation is a closely related but distinct selector, consistent with the duplication account of DART (Wen et al., 2025b); we therefore report token-set and response agreement alongside accuracy.

5.3 Cross-check on LLaVA-1.5-7B against published selectors

LLaVA-1.5 at $k = 32$ is the home setting of the published selectors, and CIG reaches the highest seven-task macro there, retaining 94.8% of the full-token score and improving over MMTok by a margin the paired interval resolves. CDPruner is the closest competitor: the 0.16 pp aggregate difference is within the paired interval, and the two methods have complementary task-level strengths—CIG ahead on SEED-Bench and GQA, CDPruner on POPE and TextVQA (Appendix J). Placed next to the exact-576 study, where the same fixed baselines trail CIG in all 24 cells, this indicates that heuristic selector rankings are regime-dependent, whereas the intervention objective carries over.

5.4 Efficiency and transfer beyond exact-576

The fixed-KV trade pays at serving-scale batching, on the backbone under memory pressure. On InternVL3-8B at batch 16, CIG is 8.6–17.9% faster end-to-end than the uncompressed model at every generation length tested, cutting peak memory from 18.6 to 15.5–16.1 GiB and raising decode throughput from 608 to 729 tokens/s at $L = 1024$. *Selection cost is included*; it is 2.2–15.4% of CIG's wall, and its falling share as L grows drives the crossover. The condition is memory pressure, not batch size: Qwen3-VL-4B never exceeds 12.1 GiB and shows no crossover, and neither backbone crosses at batch 1 or 4 (Appendix N). MMTok is often as fast or faster here, its selection being

Table 5: **Where the fixed-KV trade pays: serving-scale batching.** Median end-to-end wall (s) at batch 16 under exact-576 (20 warm-up + 200 measured requests per cell), *selection cost included* in every compressed row. The trade pays on InternVL3-8B (uncompressed peak 18.6 GiB, under memory pressure), not on Qwen3-VL-4B (never above 12.1 GiB); batch 4 and 1 do not cross on either (Appendix N).

		Wall $p50$ (s), batch 16				Δ vs Full		
Backbone	L	Full	MMTok	CIG	Batch. CIG	CIG	Batch.	Peak (GiB)
InternVL3-8B	128	3.69	2.83	3.27	2.85	**+11.3%**	**+22.6%**	18.6→15.5
	256	6.65	5.53	6.08	5.72	**+8.6%**	**+13.9%**	18.6→15.5
	512	13.04	10.79	10.71	10.96	**+17.9%**	**+16.0%**	18.6→15.6
	1024	27.85	22.42	23.12	22.74	**+17.0%**	**+18.3%**	18.6→16.1
Qwen3-VL-4B	128	3.74	3.63	3.85	3.64	-2.9%	**+2.7%**	10.6→8.8
	256	7.48	6.83	7.28	7.04	**+2.7%**	**+5.8%**	10.6→9.1
	512	14.24	13.52	14.42	14.22	-1.3%	**+0.1%**	10.9→9.7
	1024	26.86	27.44	28.54	28.93	-6.2%	-7.7%	12.1→10.8

Table 6: **Mechanism analysis on LLaVA-1.5-7B.** Left: Spearman correlation between the first-step marginal J_1 gain and each heuristic signal over the same fixed clusters, and pairwise overlap of the token sets selected at $k=32$ ($n=2000$). Right: accuracy by question type on $26{,}414$ paired FULL units at $k=32$, CIG versus MMTok, measured in a separate evaluation batch whose accuracy levels differ slightly from Table 4 (full tables: Appendices Q and R).

Signal vs J_1	ρ_s	Selection pair	Jacc.	$\lvert\cup\rvert$	Question type	n	Δ
MMTok coverage	0.081	CIG/MMTok	0.075	59.6	Counting	2,447	+6.50
CDPruner kernel	0.078	CIG/CDPruner	0.073	59.8	Fine-grained attr.	5,067	+6.32
FastV attention	0.036	MMTok/CDPruner	0.079	59.4	Spatial relation	1,973	+4.71
Cross-modal prior	0.031	*max. single-task* $\rho_s = 0.206$			OCR / text	202	+2.48
two-stage bar $\rho_s \geq 0.6$: not met					Object hallucination	14,183	+2.12
					Knowledge	2,542	+0.24

Table 7: **What the gains are attributable to.** Each component perturbed one at a time under exact-576 on InternVL3-8B and Qwen3-VL-4B ($k \in \{8, 32, 128\}$); the range spans the variants listed, each averaged over six backbone×budget cells. Per-cell values in Appendix T.

Component	Variants tested	Range of mean Δ (pp)
Objective	$J_1 \to J_2$ / J_3	-9.44 **…** -4.17
Candidate fan-out M	$M=4$ / 16 at $C=16$	-5.65 **…** $+1.27$
Cluster count C	$C=4$ / 32 at $M=8$	-2.24 **…** -1.97
Search depth	force ≥ 4 insertion rounds	-1.45
Allocator	sequential / proportional	-1.39 … -0.08
Clustering algorithm	k-means / agglomerative	$+0.30$ … $+0.50$
Member prior	random / uniform-spatial order	-0.28 … -0.06
Spatial link	enable $\gamma=0.25$	-0.04
Cluster prior	size / random ordering	$+0.00$ … $+0.00$

essentially free, so what buys throughput is *compression* and what CIG adds is the $+5.69$ pp it holds over MMTok.

Two boundaries remain. On high-resolution LLaVA-NeXT-7B at $k=128$ (1.7–2.9k AnyRes candidate tokens, baselines with model-native AnyRes adaptations), CIG reaches a four-task macro of 62.59 against 52.77 for Random and 61.23 for VisionZip but trails MMTok (64.31) and CDPruner (65.68), mostly on TextVQA; increasing C makes this worse, so large candidate pools call for an adapted allocator (Appendix L). And at batch 1 the probe prefills are not amortised, so single-request latency favours probe-free selectors (Appendix O).

6 ANALYSIS

Objective choice dominates the ablation. Replacing J_1 produces the largest accuracy drop in Table 7, far larger than changes to clustering, allocation, spatial linkage or priors; a larger candidate fan-out can help but at a substantial probe cost, and forcing extra search rounds adds probes without improving the mean. The output-fidelity objective thus contributes more than the candidate-generation heuristics that make the search tractable.

The intervention signal differs from existing proxies. Across the four heuristic families, the average Spearman correlation with the intervention score is at most $|\rho_s| = 0.081$ (Table 6; Appendix Q) and the selected sets differ substantially, so output preservation carries information that attention, similarity, diversity and coverage do not (compatible with the duplication account of DART, Wen et al., 2025b). A two-stage design in which a heuristic proposes candidates and J_1 only re-ranks them would need some heuristic to track the first-step gain closely (we set $\rho_s \geq 0.6$ as the bar); the largest single-task value is 0.206, and re-ranking by J_1 within the pooled CIG@32 $\cup$ CDPruner@32 candidates does not improve on CIG alone (72.90 versus 73.60) even though a label oracle over the same union reaches 83.90 (Appendix Q).

Gains concentrate where the answer depends on the retained pixels. On $26{,}414$ paired FULL units at $k = 32$, the margin of CIG over MMTok is largest for counting ($+6.50$ pp), fine-grained attributes ($+6.32$ pp) and spatial relations ($+4.71$ pp), moderate for object hallucination ($+2.12$ pp), and near parity for knowledge questions ($+0.24$ pp), whose answers depend least on which visual tokens survive (Table 6; per-benchmark decomposition in Appendix R, from a separate evaluation batch).

Selection geometry is content-dependent, not centre-biased. At $k = 32$ the sets CIG selects are spatially more concentrated than those of MMTok or CDPruner (normalised 6×6 entropy 0.676 versus 0.789 and 0.756) while placing more mass in the outermost patch band (0.290 versus 0.256 and 0.239); its per-task edge fraction spans the widest range (0.249 to 0.348), consistent with a policy that adapts to content rather than a fixed spatial prior, and none of the three exhibits the end-of-sequence bias reported for attention scores (Wen et al., 2025a) (Appendix S).

Oracle headroom is largely an objective gap. A label-using oracle is 11.95 pp above CIG at $k = 32$, but it optimises ground-truth probability rather than faithfulness to the full model; over the same candidate pool most of this difference remains (Appendix Q), so it is not search headroom.

7 LIMITATIONS AND DISCUSSION

CIG evaluates candidate subsets with extra model prefills, so compression does not imply lower latency in every regime: the speed-up appears when the shorter visual KV cache relieves memory pressure at serving-scale batch sizes, whereas at batch 1, and on the smaller Qwen3-VL-4B workload, selection overhead can dominate. Selections are also question-conditioned; in a preliminary reuse test a subset selected for one question lost most of its advantage on later questions about the same image. Reducing or amortising probes is therefore the main systems direction, for instance by caching the visual-prefix KV for repeated queries or by distilling the intervention signal into a lightweight scorer.

The current instantiation is also weakest on OCR-heavy and high-resolution inputs (TextVQA at $k \geq 64$, LLaVA-NeXT), and the exact-576 grid covers four mid-sized backbones and four datasets; additional architectures, high-resolution regimes and disaggregated robustness analyses are needed before treating the result as a general property of VLM compression.

8 CONCLUSION

We studied visual token selection as an output-preservation problem rather than a search for a better proxy score. CIG evaluates candidate subsets by their effect on the deployed model's predictive distribution and makes this intervention tractable with cluster-level search. Under a controlled four-backbone exact-576 protocol it is consistently strongest under aggressive compression; the ordering survives a held-out slice, free-form scoring and equal-information controls; ablations attribute most of the gain to the faithfulness objective; and serving-scale tests show the approach pays when KV memory is the binding resource.

Reproducibility statement

The selection procedure is given in full as Algorithm 1, with its termination behaviour and a worked trace in Appendix A. Appendix H lists the model identifiers, input resolutions, evaluation splits, seed and the choices fixed before the main experiments; Appendices F and I state which baselines are admissible under each protocol and how their budgets are realised. Complete per-method and per-dataset tables for the exact-576 grid and the batched-probing fidelity metrics are in Appendix B, the paired bootstrap and McNemar procedures and all intervals in Appendices C and J, the held-out and free-form evaluations in Appendices D and E, and the latency measurement protocols in Appendices N and O.

AI use statement

We used large language models to debug parts of the implementation and evaluation code and to improve the readability of the manuscript. They were not used to develop the method, design the experiments or interpret the results. All AI-assisted code and text were reviewed by the authors, who take full responsibility for the content of this paper.

References

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2503.02175.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *AAAI Conference on Artificial Intelligence*, 2025.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations (ICLR)*, 2023.

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 (ECCV)*, 2024. arXiv:2403.06764.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

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 (ICLR)*, 2026. arXiv:2508.18264.

Chenxi Du, Yongheng Deng, Jiani Liu, Yujia Zhang, Xi Chen, and Ju Ren. CoIn: Coverage and informativeness-guided token reduction for efficient large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10492–10501, 2026.

Mingjing Du, Shifei Ding, and Hongjie Jia. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems*, 99:135–145, 2016.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. ZOO-Prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39572–39582, 2026. arXiv:2509.24837.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *AAAI Conference on Artificial Intelligence*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 2024b.

Jizhihui Liu, Feiyi Du, Guangdao Zhu, Niu Lian, Jun Li, and Bin Chen. Hiprune: Training-free visual token pruning via hierarchical attention in vision-language models. *arXiv preprint arXiv:2508.00553*, 2025.

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 (ECCV)*, 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 (NeurIPS)*, 2022.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Jie Chen, Zhen Song, and Hairong Zheng. ApET: Approximation-error guided token compression for efficient VLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2602.19870.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Dhruv Parikh, Haoyang Fan, Rajgopal Kannan, and Viktor Prasanna. ConsensusDrop: Fusing visual and cross-modal saliency for efficient vision language models. *arXiv preprint arXiv:2602.00946*, 2026.

Tianfan Peng, Yuntao Du, Pengzhou Ji, Shijie Dong, Kailin Jiang, Mingchuan Ma, Yijun Tian, Jinhe Bi, Qian Li, Wei Du, Feng Xiao, and Lizhen Cui. Can visual input be compressed? a visual token compression benchmark for large multimodal models. *arXiv preprint arXiv:2511.02650*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2403.15388.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. A survey of token compression for efficient multimodal large language models. *Transactions on Machine Learning Research (TMLR)*, 2026. arXiv:2507.20198.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Huanyu Wang, Jushi Kai, Haoli Bai, Lu Hou, Bo Jiang, Ziwei He, and Zhouhan Lin. Fourier-VLM: Compressing vision tokens in the frequency domain for large vision-language models. *arXiv preprint arXiv:2508.06038*, 2025a.

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

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

Yi Wang, Haofei Zhang, Qihan Huang, Anda Cao, Gongfan Fang, Wei Wang, Xuan Jin, Jie Song, Mingli Song, and Xinchao Wang. Rethinking token reduction for large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24727–24737, 2026. arXiv:2603.21701.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15537–15549, 2025a. doi: 10.18653/v1/2025.findings-acl.802.

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 (EMNLP)*, pp. 9961–9980, 2025b. doi: 10.18653/v1/2025.emnlp-main.505.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2410.17247.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2412.04467.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *AAAI Conference on Artificial Intelligence*, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Accelerating the development of large multimodal models. `https://github.com/EvolvingLMMs-Lab/lmms-eval`, 2024a.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minghua Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [CLS] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818v1*, 2024b.

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/CVF International Conference on Computer Vision (ICCV)*, 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 (NeurIPS)*, 2025b. arXiv:2506.10967.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis 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 (ICML)*, 2025c. arXiv:2410.04417.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Wang, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Appendix

A Selection algorithm: termination and worked trace

Algorithm 1 gives the complete procedure; this appendix covers its termination behaviour and a worked trace.

Termination. ALLOC emits tokens only from the selected clusters and returns $\min(k, \sum_{c\in\mathcal{C}} |c|)$ indices, so it underfills whenever the selected clusters are collectively smaller than k and the outer loop continues. The procedure stops at iteration $T = \min\{t : \sum_{i\le t} |c_i| \ge k\}$ for the chosen sequence $c_1, c_2, \ldots$; every emitted token belongs to a probed cluster, and if all C clusters are selected with $\sum_c |c| < k$ (possible only when $k > N$) the whole sequence is returned.

Worked trace. Table 8 runs the deployed allocator on two illustrative eight-cluster size profiles, with the highest-prior cluster large (A) or small (B); the two rows bracket the behaviour observed in practice.

In profile A at $k \le 128$ the procedure is a one-step selection: it scores $M = 8$ single-cluster subsets and keeps the best, so the gain-density normaliser of Equation 4 is inactive (all candidates add the

Table 8: Search depth of Algorithm 1 for $C = 8$, $N = 576$, $M = 8$. Profile A has cluster sizes $(138, 96, 81, 74, 63, 51, 42, 31)$ in decreasing cluster-prior order; profile B has the same multiset in increasing order. *Probes* counts subset prefills, excluding the two reference reads and the final re-read.

Profile	k	Iterations	Trace of $\lvert S_t \rvert$ (clusters admitted)
A: leading cluster large	8	1	8 (1 cluster; 8 probes)
	16	1	16 (1 cluster; 8 probes)
	32	1	32 (1 cluster; 8 probes)
	64	1	64 (1 cluster; 8 probes)
	128	1	128 (1 cluster; 8 probes)
	192	2	$138 \to 192$ (2 clusters; 15 probes)
B: leading cluster small	8	1	8 (1 cluster; 8 probes)
	16	1	16 (1 cluster; 8 probes)
	32	2	$31 \to 32$ (2 clusters; 15 probes)
	64	2	$31 \to 64$ (2 clusters; 15 probes)
	128	4	$31 \to 73 \to 124 \to 128$ (4 clusters; 26 probes)
	192	5	$31 \to 73 \to 124 \to 187 \to 192$ (5 clusters; 30 probes)

same number of tokens). Profile B, or a large k, produces multi-step insertion. Both regimes occur in our runs: the mean probe volume at $k = 32$ (about 400 candidate tokens per request, Appendix O) exceeds a single eight-probe round of ≤ 32-token subsets. Deeper search (finer clusters with more insertion steps, or beam search over cluster combinations) would test whether the objective or the candidate generator is the binding constraint, and is left to future work.

B EXACT-576 GRID: COMPLETE TABLES

Protocol: Full=576 native visual tokens on every backbone, budgets $k \in \{8, 16, 32, 64, 128, 192\}$, four datasets at 500 examples each, BF16, seed 42, tokens deleted and positional encodings repaired. Each backbone contributes $12{,}000$ sample–budget records, $48{,}000$ in total; the full-token condition reproduces the unmodified model.

Table 9: Four-task macro accuracy (%) per method under exact-576; CIG rows are bold where they exceed every heuristic in the same column, which is every cell.

Backbone	Method	k=8	k=16	k=32	k=64	k=128	k=192
InternVL3-8B	Full	87.65	87.65	87.65	87.65	87.65	87.65
	Random	68.00	72.70	77.45	81.00	83.85	85.10
	MMTok	67.75	73.85	79.45	83.05	86.05	86.55
	DivPrune	67.15	72.35	78.00	82.60	84.80	86.10
	CDPruner	70.80	74.45	78.80	82.20	84.40	85.20
	CIG	**81.80**	**83.55**	**85.40**	**86.25**	**86.40**	**87.30**
	Batched CIG	82.15	83.25	84.75	86.05	86.35	86.95
Qwen2.5-VL-7B	Full	85.90	85.90	85.90	85.90	85.90	85.90
	Random	66.95	71.65	76.30	78.35	82.45	83.20
	MMTok	68.75	72.90	76.75	81.30	83.05	84.20
	DivPrune	63.70	70.45	75.55	80.00	82.65	84.05
	CDPruner	60.80	66.65	71.35	76.50	80.05	82.35
	CIG	**80.55**	**82.35**	**84.20**	**84.90**	**85.30**	**85.75**
	Batched CIG	80.20	82.25	84.00	85.10	85.15	85.85
InternVL3.5-4B	Full	84.20	84.20	84.20	84.20	84.20	84.20
	Random	65.75	69.00	73.50	76.20	79.75	80.85
	MMTok	65.00	71.70	73.90	77.25	79.30	82.20
	DivPrune	62.70	67.80	73.35	77.25	80.00	81.15
	CDPruner	63.80	67.80	71.50	76.10	79.70	81.30
	CIG	**78.65**	**80.05**	**80.75**	**81.50**	**82.50**	**82.45**
	Batched CIG	78.55	80.15	80.75	81.85	82.50	82.20
Qwen3-VL-4B	Full	87.95	87.95	87.95	87.95	87.95	87.95
	Random	66.80	71.50	75.65	80.25	84.10	85.35
	MMTok	71.65	76.30	80.40	83.60	85.40	86.00
	DivPrune	65.75	71.85	77.45	82.50	84.30	85.40
	CDPruner	62.95	67.35	73.05	77.35	82.60	84.55
	CIG	**82.25**	**84.15**	**86.05**	**86.55**	**86.90**	**87.40**
	Batched CIG	82.45	84.10	85.95	86.60	87.20	87.50

Table 10: Fidelity of Batched CIG against the exact sequential selector, per budget (summarised as ranges in Table 3). Gate: token Jaccard ≥ 0.8 *and* response match ≥ 0.98; Δacc is batched minus exact four-task macro (pp). All six InternVL3-8B rows fail the gate while their accuracy is preserved to within 0.65 pp: the gate is sufficient for equivalence, not necessary.

Backbone	k	Jaccard	Resp. match	Δacc	Gate	Backbone	k	Jaccard	Resp. match	Δacc	Gate
InternVL3-8B	8	0.7622	0.9695	$+0.35$	fail	Qwen2.5-VL-7B	8	0.8718	0.9910	-0.35	**pass**
	16	0.7586	0.9715	-0.30	fail		16	0.8874	0.9935	-0.10	**pass**
	32	0.7317	0.9750	-0.65	fail		32	0.8791	0.9915	-0.20	**pass**
	64	0.6972	0.9745	-0.20	fail		64	0.8543	0.9905	$+0.20$	**pass**
	128	0.6637	0.9830	-0.05	fail		128	0.8192	0.9890	-0.15	**pass**
	192	0.6708	0.9865	-0.35	fail		192	0.8026	0.9925	$+0.10$	**pass**
InternVL3.5-4B	8	0.9142	0.9920	-0.10	**pass**	Qwen3-VL-4B	8	0.9232	0.9925	$+0.20$	**pass**
	16	0.9104	0.9905	$+0.10$	**pass**		16	0.9085	0.9945	-0.05	**pass**
	32	0.8927	0.9885	$+0.00$	**pass**		32	0.8949	0.9935	-0.10	**pass**
	64	0.8713	0.9860	$+0.35$	**pass**		64	0.8868	0.9950	$+0.05$	**pass**
	128	0.8300	0.9865	$+0.00$	**pass**		128	0.8551	0.9925	$+0.30$	**pass**
	192	0.8221	0.9850	-0.25	**pass**		192	0.8452	0.9950	$+0.10$	**pass**

Table 11: Per-dataset accuracy (%) under the exact-576 protocol for all four backbones. MMBench: MMBench-EN-dev; SEED: SEED-Bench-image; SQA: ScienceQA-IMG.

		InternVL3-8B						**Qwen2.5-VL-7B**					
Dataset	Method	k=8	16	32	64	128	192	k=8	16	32	64	128	192
POPE	Full	89.60	89.60	89.60	89.60	89.60	89.60	88.00	88.00	88.00	88.00	88.00	88.00
	Random	70.60	76.40	80.60	82.20	87.00	87.40	66.60	72.20	78.20	80.60	85.20	85.00
	MMTok	71.60	80.20	83.80	87.00	88.60	88.00	69.60	78.00	81.40	85.00	87.60	88.20
	DivPrune	71.20	77.60	81.60	85.60	88.00	87.80	65.80	74.00	80.20	83.80	86.00	87.40
	CDPruner	75.00	78.60	80.60	84.60	86.00	87.40	60.80	69.00	75.20	80.40	83.80	85.40
	CIG	87.60	87.80	88.20	88.20	88.60	89.20	82.20	85.20	86.40	87.20	88.20	88.80
	Batched CIG	88.40	87.60	88.20	88.40	88.40	88.60	82.00	85.40	86.40	87.80	88.20	88.60
MMBench	Full	87.80	87.80	87.80	87.80	87.80	87.80	88.80	88.80	88.80	88.80	88.80	88.80
	Random	70.40	76.00	82.80	84.00	85.40	86.80	69.80	74.20	78.60	82.00	84.60	86.20
	MMTok	70.40	76.40	81.80	85.40	87.20	87.60	72.80	75.80	80.00	84.00	84.60	86.00
	DivPrune	69.20	76.60	81.80	84.60	85.80	87.00	64.60	73.00	78.20	83.20	85.20	86.80
	CDPruner	74.20	78.80	84.20	85.80	87.00	87.00	62.20	68.20	73.80	78.60	84.00	85.60
	CIG	81.40	83.80	86.60	87.60	87.00	88.20	82.80	84.00	86.00	86.80	87.80	87.00
	Batched CIG	81.20	83.40	85.20	86.60	87.60	88.20	82.60	83.80	85.60	86.80	87.40	87.40
SEED	Full	76.40	76.40	76.40	76.40	76.40	76.40	76.80	76.80	76.80	76.80	76.80	76.80
	Random	55.60	59.80	63.00	70.00	71.00	71.40	54.20	61.80	67.80	68.60	74.20	74.20
	MMTok	52.40	59.80	68.20	71.40	74.40	74.60	53.80	59.20	65.20	71.00	72.40	73.60
	DivPrune	50.80	55.00	62.40	70.40	73.40	74.00	49.20	57.00	63.60	69.00	73.40	74.80
	CDPruner	54.20	58.80	65.80	70.20	71.60	73.00	46.20	52.80	58.60	65.00	69.60	73.40
	CIG	69.40	71.40	73.40	74.00	74.80	75.80	70.80	72.60	75.20	75.60	76.20	78.40
	Batched CIG	69.80	71.00	72.40	74.20	74.00	75.00	70.40	72.80	75.00	76.20	76.00	78.80
SQA	Full	96.80	96.80	96.80	96.80	96.80	96.80	90.00	90.00	90.00	90.00	90.00	90.00
	Random	75.40	78.60	83.40	87.80	92.00	94.80	77.20	78.40	80.60	82.20	85.80	87.40
	MMTok	76.60	79.00	84.00	88.40	94.00	96.00	78.80	78.60	80.40	85.20	87.60	89.00
	DivPrune	77.40	80.20	86.20	89.80	92.00	95.60	75.20	77.80	80.20	84.00	86.00	87.20
	CDPruner	79.80	81.60	84.60	88.20	93.00	93.40	74.00	76.60	77.80	82.00	82.80	85.00
	CIG	88.80	91.20	93.40	95.20	95.20	96.00	86.40	87.60	89.20	90.00	89.00	88.80
	Batched CIG	89.20	91.00	93.20	95.00	95.40	96.00	85.80	87.00	89.00	89.60	89.00	88.60

		InternVL3.5-4B						**Qwen3-VL-4B**					
Dataset	Method	k=8	16	32	64	128	192	k=8	16	32	64	128	192
POPE	Full	82.60	82.60	82.60	82.60	82.60	82.60	89.60	89.60	89.60	89.60	89.60	89.60
	Random	67.60	72.80	77.60	79.80	81.20	82.60	65.60	73.00	76.80	81.20	86.80	87.60
	MMTok	66.40	75.20	76.40	80.40	80.80	81.40	76.60	81.40	84.40	89.20	89.40	90.80
	DivPrune	65.80	73.80	79.00	81.20	80.80	81.40	66.80	74.80	82.80	89.60	88.80	91.00
	CDPruner	67.20	73.60	75.20	78.40	79.60	80.40	62.00	70.40	75.80	80.00	85.80	88.40
	CIG	82.60	82.80	83.00	82.60	83.20	81.40	88.00	89.00	88.80	89.40	89.60	90.20
	Batched CIG	82.60	83.20	83.20	84.00	82.40	81.40	88.00	88.20	88.60	89.80	89.80	90.20
MMBench	Full	84.40	84.40	84.40	84.40	84.40	84.40	90.60	90.60	90.60	90.60	90.60	90.60
	Random	66.80	72.20	77.40	78.40	82.20	82.80	71.00	77.00	81.80	84.40	86.40	88.40
	MMTok	67.60	74.20	77.00	80.40	82.40	84.20	74.20	81.00	84.20	87.80	88.80	87.80
	DivPrune	66.40	72.20	75.80	81.20	83.00	84.00	71.00	77.20	82.60	85.60	87.40	87.20
	CDPruner	66.40	70.80	75.80	80.40	82.80	82.60	69.40	75.40	79.00	83.40	86.80	87.20
	CIG	78.40	79.40	81.60	82.60	83.20	83.60	85.40	87.40	89.80	89.40	89.60	90.20
	Batched CIG	78.20	79.60	81.20	82.40	83.20	82.80	85.80	87.60	89.60	89.60	90.20	90.40
SEED	Full	75.20	75.20	75.20	75.20	75.20	75.20	76.40	76.40	76.40	76.40	76.40	76.40
	Random	53.20	56.80	63.00	68.80	71.60	70.40	55.40	59.40	63.00	70.60	73.20	73.40
	MMTok	51.20	60.60	63.60	65.20	68.40	73.00	59.80	64.20	69.40	71.60	72.40	73.60
	DivPrune	44.80	49.60	57.60	65.60	70.80	72.80	50.40	56.40	62.60	69.00	72.40	73.40
	CDPruner	46.80	50.60	57.20	63.60	71.20	71.60	44.80	48.20	55.60	62.00	69.80	71.80
	CIG	68.00	70.60	72.20	73.00	73.60	74.20	70.40	73.40	74.60	74.60	75.00	75.40
	Batched CIG	67.60	70.40	72.20	72.80	73.80	73.80	70.60	73.60	74.80	74.40	75.00	75.40
SQA	Full	94.60	94.60	94.60	94.60	94.60	94.60	95.20	95.20	95.20	95.20	95.20	95.20
	Random	75.40	74.20	76.00	77.80	84.00	87.60	75.20	76.60	81.00	84.80	90.00	92.00
	MMTok	74.80	76.80	78.60	83.00	85.60	90.20	76.00	78.60	83.60	85.80	91.00	91.80
	DivPrune	73.80	75.60	81.00	81.00	85.40	86.40	74.80	79.00	81.80	85.80	88.60	90.00
	CDPruner	74.80	76.20	77.80	82.00	85.20	90.60	75.60	75.40	81.80	84.00	88.00	90.80
	CIG	85.60	87.40	86.20	87.80	90.00	90.60	85.20	86.80	91.00	92.80	93.40	93.80
	Batched CIG	85.80	87.40	86.40	88.20	90.60	90.80	85.40	87.00	90.80	92.60	93.80	94.00

C UNCERTAINTY FOR THE EXACT-576 GRID

Each exact-576 cell aggregates $4 \times 500 = 2{,}000$ paired units, one per item and task. The bootstrap is stratified within task, resampling items and then weighting the four tasks equally to match the macro definition, with 10,000 draws. For the *best heuristic* opponent the strongest baseline is re-selected inside every resample, so that interval absorbs the within-cell choice and is deliberately wider than any fixed-opponent interval. McNemar is the exact two-sided binomial test on discordant pairs.

Table 12: Paired 95% bootstrap intervals, CIG minus each opponent. **Bold** where the interval excludes zero. The Random column is omitted for width; it excludes zero in all 24 cells and is summarised in the next table.

		MMTok		DivPrune		CDPruner		best heur.	
Backbone	k	Δ	95% CI	Δ	95% CI	Δ	95% CI	Δ	95% CI
InternVL3-8B	8	**+14.05**	[+12.0, +16.1]	**+14.65**	[+12.7, +16.6]	**+11.00**	[+9.1, +12.9]	**+11.00**	[+8.5, +13.4]
	16	**+9.70**	[+7.9, +11.5]	**+11.20**	[+9.4, +13.0]	**+9.10**	[+7.3, +10.9]	**+9.10**	[+6.6, +11.0]
	32	**+5.95**	[+4.5, +7.6]	**+7.40**	[+5.8, +9.0]	**+6.60**	[+5.1, +8.1]	**+5.95**	[+3.6, +7.7]
	64	**+3.20**	[+1.9, +4.5]	**+3.65**	[+2.3, +5.1]	**+4.05**	[+2.7, +5.5]	**+3.20**	[+0.8, +4.7]
	128	+0.35	[−0.6, +1.4]	**+1.60**	[+0.5, +2.7]	**+2.00**	[+0.9, +3.1]	+0.35	[−1.8, +2.2]
	192	+0.75	[−0.1, +1.7]	**+1.20**	[+0.2, +2.2]	**+2.10**	[+1.1, +3.1]	+0.75	[−1.4, +2.3]
Qwen2.5-VL-7B	8	**+11.80**	[+10.0, +13.6]	**+16.85**	[+14.9, +18.8]	**+19.75**	[+17.8, +21.8]	**+11.80**	[+9.2, +14.2]
	16	**+9.45**	[+7.8, +11.1]	**+11.90**	[+10.1, +13.7]	**+15.70**	[+13.8, +17.6]	**+9.45**	[+6.9, +11.6]
	32	**+7.45**	[+6.0, +8.9]	**+8.65**	[+7.1, +10.3]	**+12.85**	[+11.1, +14.7]	**+7.45**	[+4.9, +9.2]
	64	**+3.60**	[+2.4, +4.9]	**+4.90**	[+3.6, +6.2]	**+8.40**	[+6.8, +10.1]	**+3.60**	[+1.3, +5.6]
	128	**+2.25**	[+1.2, +3.4]	**+2.65**	[+1.6, +3.8]	**+5.25**	[+3.8, +6.7]	+2.25	[−0.2, +3.7]
	192	**+1.55**	[+0.6, +2.5]	**+1.70**	[+0.7, +2.7]	**+3.40**	[+2.1, +4.8]	+1.55	[−0.9, +3.0]
InternVL3.5-4B	8	**+13.65**	[+11.8, +15.6]	**+15.95**	[+13.9, +18.0]	**+14.85**	[+12.8, +16.8]	**+12.90**	[+10.1, +15.0]
	16	**+8.35**	[+6.6, +10.2]	**+12.25**	[+10.4, +14.2]	**+12.25**	[+10.3, +14.2]	**+8.35**	[+5.8, +10.9]
	32	**+6.85**	[+5.2, +8.5]	**+7.40**	[+5.7, +9.1]	**+9.25**	[+7.5, +11.0]	**+6.85**	[+4.0, +8.6]
	64	**+4.25**	[+2.7, +5.8]	**+4.25**	[+2.7, +5.8]	**+5.40**	[+3.8, +7.0]	**+4.25**	[+1.4, +5.8]
	128	**+3.20**	[+1.8, +4.6]	**+2.50**	[+1.1, +3.9]	**+2.80**	[+1.4, +4.2]	+2.50	[−0.3, +3.9]
	192	+0.25	[−1.0, +1.5]	+1.30	[+0.0, +2.6]	+1.15	[−0.1, +2.4]	+0.25	[−2.2, +2.0]
Qwen3-VL-4B	8	**+10.60**	[+8.9, +12.3]	**+16.50**	[+14.6, +18.4]	**+19.30**	[+17.3, +21.2]	**+10.60**	[+8.0, +13.2]
	16	**+7.85**	[+6.3, +9.5]	**+12.30**	[+10.6, +14.1]	**+16.80**	[+14.8, +18.8]	**+7.85**	[+5.4, +10.2]
	32	**+5.65**	[+4.2, +7.1]	**+8.60**	[+7.0, +10.2]	**+13.00**	[+11.2, +14.8]	**+5.65**	[+3.4, +7.9]
	64	**+2.95**	[+1.6, +4.3]	**+4.05**	[+2.7, +5.5]	**+9.20**	[+7.6, +10.8]	**+2.95**	[+0.8, +4.9]
	128	**+1.50**	[+0.4, +2.6]	**+2.60**	[+1.4, +3.8]	**+4.30**	[+3.0, +5.5]	+1.50	[−0.6, +3.2]
	192	**+1.40**	[+0.5, +2.4]	**+2.00**	[+1.0, +3.0]	**+2.85**	[+1.7, +4.0]	+1.40	[−0.8, +2.9]

Table 13: Aggregates over all 24 cells against each *fixed* opponent. The cluster bootstrap resamples (backbone, task) units; an equal-family stratified paired Wald interval is reported as a secondary check.

Opponent	Cells won	Mean Δ	Cluster-bootstrap 95% CI	Wald CI	Cells containing 0
vs Random	24 / 24	+7.39	[+6.42, +8.36]	[+7.06, +7.71]	none
vs MMTok	24 / 24	+5.69	[+4.78, +6.61]	[+5.39, +5.99]	IV3-8B k128/192, IV3.5-4B k192
vs DivPrune	24 / 24	+7.34	[+6.19, +8.57]	[+7.01, +7.66]	IV3.5-4B k192
vs CDPruner	24 / 24	+8.81	[+7.54, +10.11]	[+8.47, +9.14]	IV3.5-4B k192

Sign tests. Treating the 24 cells as independent gives a two-sided exact binomial $p \approx 1.2 \times 10^{-7}$, but the cells share datasets, prompts, full-model predictions and neighbouring budgets. Clustering the unit of analysis at the six budgets gives $p = 0.031$ and at the four backbones $p = 0.125$. The naive value is anti-conservative, so the claim rests on the per-cell intervals and the cluster bootstrap above.

Effect of pairing. Against the per-cell best heuristic, 16 of 24 intervals exclude zero: pairing tightens each interval, but re-selecting the opponent inside every resample widens this particular one, and the second effect dominates at $k \geq 128$. Against any *fixed* opponent, 21 to 24 of 24 intervals exclude zero, and all four aggregate intervals do.

D HELD-OUT SLICE

The exact-576 grid of Table 1 is evaluated on the first 500 items of each task, which are also the items of the LLaVA-1.5 SCREEN on which every hyper-parameter was fixed. To rule out an item-selection

effect we re-ran the grid on items 501–1,000 of each task (disjoint from SCREEN) for InternVL3-8B and Qwen2.5-VL-7B at $k \in \{32, 64\}$, with an otherwise unchanged pipeline and readout. Table 14 shows that CIG leads every heuristic in every task cell and that the pooled macro margin over the best heuristic ($+4.05$ and $+4.23$ pp) matches the SCREEN margin pooled over the same budgets ($+4.58$ and $+5.53$ pp). The free-form generation results of Appendix E are reported on the same held-out slice.

Table 14: Held-out check of the exact-576 grid on items 501–1,000 of each task (disjoint from every item used for configuration), accuracy (%) pooled over $k \in \{32, 64\}$ (1,000 paired records per cell), first-token readout, true deletion. CIG rows are bold where they exceed every heuristic in the column, which is every cell.

	InternVL3-8B					Qwen2.5-VL-7B				
Method	POPE	MMBench	SEED	SQA	Macro	POPE	MMBench	SEED	SQA	Macro
Full	86.6	89.8	78.2	95.2	87.4	88.4	87.8	81.2	86.0	85.8
Random	82.2	83.7	69.3	87.0	80.5	80.0	83.9	69.2	80.9	78.5
MMTok	83.9	84.8	69.9	87.2	81.4	83.6	85.4	71.2	81.2	80.3
DivPrune	85.5	83.0	67.7	87.8	81.0	82.1	83.4	66.4	81.3	78.3
CDPruner	84.4	84.4	66.6	86.9	80.6	78.9	81.3	59.9	78.2	74.6
CIG	**85.9**	**87.3**	**75.1**	**93.7**	**85.5**	**86.3**	**87.7**	**78.1**	**86.2**	**84.6**
Batched CIG	86.0	88.3	76.0	94.2	86.1	86.3	87.3	78.5	86.5	84.7

For the LLaVA-1.5 protocol, whose FULL splits contain the SCREEN items, Table 15 recomputes the $k = 32$ comparison of Table 4 on the items with index ≥ 500 only, from the per-item records. The per-task values move by at most 0.55 pp relative to the FULL splits, so the overlap has no measurable effect.

Table 15: LLaVA-1.5-7B at $k = 32$ on FULL minus SCREEN (items with index ≥ 500), recomputed from the per-item records of Table 4; accuracy (%) and paired bootstrap 95% intervals (10,000 draws) for CIG minus each opponent. MMBench, whose circular-variant score is not defined per item, is omitted.

Task	Full	MMTok	CDPruner	CIG	CIG−MMTok	CIG−CDPruner
POPE (n=8,500)	86.94	85.14	87.67	85.67	$+0.53$ $[-0.07, +1.13]$	-2.00 $[-2.65, -1.38]$
SEED-Bench (n=13,733)	66.13	59.70	60.78	64.36	$+4.65$ $[+4.00, +5.32]$	$+3.58$ $[+2.94, +4.21]$
ScienceQA (n=1,517)	70.07	69.48	68.09	69.87	$+0.40$ $[-1.12, +1.91]$	$+1.78$ $[+0.13, +3.36]$

E FREE-FORM GENERATION ON THE NATIVE BACKBONES

The exact-576 grid scores answers with a first-token readout. Here every method's output is instead generated greedily (`max_new_tokens`$= 16$) and scored with the `lmms-eval` metrics for GQA testdev-balanced (exact match after normalisation) and TextVQA-val (VQA accuracy with fractional credit), on both the SCREEN and the held-out slice, for all four backbones and $k \in \{32, 64, 128\}$ (Tables 16 and 17). For open-ended tasks CIG uses the answer support described in Section 3. Two label-free quantities accompany each cell: the fraction of items whose generated answer equals the full model's own answer (fidelity to Full), and the selection time per item (single RTX 5090; 0.23–0.73 s for CIG against 0.02–0.06 s for the heuristics).

Three regularities hold on both slices. At $k = 32$, CIG is the best selector in all 16 backbone–task cells, and its paired interval against MMTok excludes zero in 5 of 8 SCREEN and 6 of 8 held-out cells. At $k = 64$ the two are statistically indistinguishable on GQA and MMTok moves ahead on TextVQA (significantly only on Qwen3-VL-4B). At $k = 128$ MMTok leads TextVQA on every backbone, significantly on three of four, while GQA remains a tie; this is the OCR boundary of Section 5.4, which the readout grid places at the same budgets. Fidelity to the full model's answer is highest for CIG in every cell at $k = 32$, as the objective predicts; the fidelity of Batched CIG coincides with that of the sequential selector on GQA and differs from it by at most 0.6 pp on TextVQA.

Table 16: Free-form generation on the SCREEN slice (items 1–500), exact-576, greedy decoding scored with the `lmms-eval` GQA and TextVQA metrics ($n = 500$ per cell). Best pruned method per column in bold. Fidelity to Full is the fraction of items whose generated answer equals the full model's own answer.

		GQA exact match (%)			TextVQA VQA acc. (%)			Fidelity to Full, k=32 (%)	
Backbone	Method	k=32	64	128	k=32	64	128	GQA	TextVQA
InternVL3-8B	Full		59.4			69.5		100.0	100.0
	Random	46.4	52.2	56.0	23.7	32.9	47.2	62.8	23.0
	MMTok	53.2	**58.2**	**60.2**	40.4	**54.5**	**64.2**	69.8	40.8
	DivPrune	52.0	57.2	59.4	38.9	47.5	55.1	66.0	40.6
	CDPruner	53.4	56.2	57.4	24.9	34.4	46.1	67.0	26.2
	CIG	**56.4**	54.6	55.2	**44.8**	51.0	56.5	72.4	48.8
	Batched CIG	56.4	54.6	55.2	44.6	52.7	57.0	72.4	48.2
Qwen2.5-VL-7B	Full		58.2			82.5		100.0	100.0
	Random	51.2	53.8	55.8	34.8	48.5	62.1	71.0	33.4
	MMTok	53.8	56.8	58.4	52.5	**68.4**	**75.1**	76.2	50.0
	DivPrune	52.2	55.4	**59.0**	43.8	61.8	69.8	70.6	43.6
	CDPruner	47.4	52.4	56.6	30.3	41.9	54.5	62.4	30.0
	CIG	**57.8**	**58.0**	58.2	**60.0**	66.0	72.1	84.8	62.0
	Batched CIG	57.8	58.0	58.2	60.1	66.8	72.2	84.8	62.2
InternVL3.5-4B	Full		55.4			64.4		100.0	100.0
	Random	47.8	51.2	53.8	17.8	23.8	33.7	59.2	19.0
	MMTok	48.2	52.6	52.6	32.7	**43.7**	**53.8**	59.4	34.4
	DivPrune	46.4	51.4	54.2	31.1	38.6	46.4	58.6	34.2
	CDPruner	46.0	48.2	51.6	22.6	30.1	42.5	53.8	27.2
	CIG	**54.4**	**54.8**	**54.6**	**35.0**	41.0	47.9	76.4	41.6
	Batched CIG	54.4	54.8	54.6	35.3	41.2	48.0	76.4	42.0
Qwen3-VL-4B	Full		63.2			80.2		100.0	100.0
	Random	52.6	57.4	58.2	31.0	40.6	58.4	69.2	29.6
	MMTok	57.8	59.4	60.2	51.6	**66.2**	**74.6**	78.8	52.4
	DivPrune	53.2	58.0	61.2	47.6	59.4	71.1	72.6	47.2
	CDPruner	52.4	55.8	57.8	30.9	47.2	60.8	64.2	32.8
	CIG	**61.8**	**60.8**	**62.2**	**53.4**	59.9	65.2	88.4	57.2
	Batched CIG	61.8	60.8	62.2	53.5	59.8	64.8	88.4	57.0

Table 17: Free-form generation on the held-out slice (items 501–1,000); columns as in Table 16. Table 2 in the main text shows the $k = 32$ columns of this table.

		GQA exact match (%)			TextVQA VQA acc. (%)			Fidelity to Full, k=32 (%)	
Backbone	Method	k=32	64	128	k=32	64	128	GQA	TextVQA
InternVL3-8B	Full		63.8			70.9		100.0	100.0
	Random	49.6	55.4	60.0	25.2	34.9	46.9	64.6	24.2
	MMTok	56.2	59.0	62.8	39.6	**54.1**	**66.1**	68.8	38.8
	DivPrune	54.8	59.0	62.6	41.3	48.8	57.9	69.4	43.8
	CDPruner	53.4	60.0	59.4	25.4	36.6	47.2	71.4	26.6
	CIG	**58.4**	**60.2**	**63.8**	**45.8**	53.7	58.7	78.0	48.6
	Batched CIG	58.4	60.2	63.8	45.1	52.4	58.5	78.0	48.0
Qwen2.5-VL-7B	Full		66.0			83.3		100.0	100.0
	Random	52.0	58.4	61.0	35.1	49.6	62.1	66.6	33.0
	MMTok	58.0	**63.8**	61.4	51.0	65.6	**77.4**	73.4	48.6
	DivPrune	54.8	60.0	63.0	45.9	61.2	72.3	70.2	45.0
	CDPruner	48.8	54.4	59.2	31.3	41.9	54.3	58.8	29.6
	CIG	**62.0**	63.0	**64.2**	**61.6**	**67.4**	71.2	85.4	62.8
	Batched CIG	62.0	63.0	64.2	61.5	67.9	71.2	85.4	62.4
InternVL3.5-4B	Full		59.4			65.0		100.0	100.0
	Random	49.4	52.4	57.2	21.4	26.9	35.1	58.0	22.2
	MMTok	51.6	56.4	58.8	32.8	43.4	**56.0**	65.0	34.2
	DivPrune	51.6	53.4	57.8	34.9	42.6	48.8	66.4	35.6
	CDPruner	47.0	51.6	55.0	26.5	32.9	43.9	58.8	27.0
	CIG	**56.2**	**58.4**	**59.6**	**36.8**	**44.5**	51.0	78.4	39.4
	Batched CIG	56.2	58.4	59.6	36.6	44.1	50.2	78.4	39.2
Qwen3-VL-4B	Full		65.6			78.5		100.0	100.0
	Random	55.8	58.0	62.4	30.4	40.2	53.3	67.8	28.2
	MMTok	58.4	61.6	64.0	49.9	**67.8**	**75.6**	77.4	51.6
	DivPrune	56.0	61.2	**64.6**	47.2	59.5	70.0	68.4	45.4
	CDPruner	48.8	52.8	59.8	34.1	51.2	63.5	62.4	34.4
	CIG	**63.8**	**63.8**	64.0	**53.0**	58.9	64.7	87.0	54.2
	Batched CIG	63.8	63.8	64.0	53.3	59.0	64.9	87.0	54.4

F BASELINE COMPARABILITY UNDER EXACT-576

Six of the ten selectors we integrated transfer to the exact-576 native protocol; four do not, for the reasons given below. We do not port approximations of these methods onto architectures that lack their defining signal.

Table 18: Admissibility of each method under the exact-576 native protocol on InternVL3-8B, Qwen2.5-VL-7B, InternVL3.5-4B and Qwen3-VL-4B. NA denotes a formal incomparability or unreachability, not a missing experiment.

Method	Status	Basis
Random	PASS	deterministic index selection; architecture-independent
MMTok	PASS	native projected-token coverage selection
DivPrune	PASS	feature-level max–min diversity
CDPruner	PASS	feature-level conditional DPP on image/text embeddings
CIG	PASS	exact intervention search with native token deletion
Batched CIG	PASS	batched probes; fidelity reported separately
FastV	NA	official $K=2$ prunes on decoder layer-2 attention, which needs a language-model layer hook and cannot be represented by pre-LLM deletion; its schedule also has a layer-average floor at low budgets
VisionZip	NA	requires CLIP `[CLS]` attention, a key metric and a contextual merge; the InternVL/Qwen vision towers expose no verifiable CLIP-equivalent signal
SparseVLM	NA	depends on its released per-layer sparsity schedule; no architecture adaptation exists for these backbones
PyramidDrop	NA	fixed stage schedule cannot realise the requested budgets at $N=576$, and no equivalent stage implementation exists here

The four NA methods are all evaluated in their published settings on LLaVA-1.5-7B and LLaVA-NeXT-7B (Table 4, Appendices I and L), so every method appears in at least one protocol; what changes across protocols is which methods are *comparable*.

G EQUAL-INFORMATION CONTROLS

The published heuristics compared in Table 1 operate on encoder or projector features and never see the model's output. CIG does: it runs one 576-token reference prefill and reads $p_\star$. To separate the value of this information from the value of the intervention search, three label-free selectors were given the same reference prefill and evaluated under the exact-576 protocol on all four backbones, four tasks, 500 SCREEN items and six budgets, with one signal per item reused across budgets:

- GRADFULL: the input$\times$gradient norm, per visual token, of the log-probability of the full model's own argmax answer token, back-propagated from the reference prefill to the projected visual embeddings; top-k by norm.
- IG8: eight-step integrated gradients of the same quantity from an all-zero visual baseline (eight forward and backward passes, comparable to CIG's probe budget).
- KNOCKOUT: CIG's own DPC-kNN clusters ($C=8$), each scored by $\mathrm{KL}(p_\star \,\|\, p_{\setminus c})$ under true deletion of that cluster from the full context (eight probes, the first-round budget of CIG), then allocated with the same round-robin ALLOC.

ZOO-Prune (Kim et al., 2026) is not included because its released implementation has no true-deletion adapter for these backbones (cf. Appendix F).

Table 19 gives the results. GRADFULL exceeds the best input-level heuristic in all 24 cells ($+0.45$ to $+7.45$ pp), so access to the output distribution, not the search, is what the heuristics lack at moderate budgets. CIG exceeds GRADFULL in all 16 cells with $k \le 64$ ($+0.10$ to $+10.95$ pp; $+5.2$ to $+11.0$ pp at $k=8$) and is within 1.2 pp of it at $k \ge 128$, where all output-informed selectors sit within about one point of the full model. IG8 costs eight backward passes and is not better than GRADFULL at $k \le 32$; KNOCKOUT trails CIG at every budget ($+0.25$ to $+12.55$ pp), reproducing on the native backbones the gain-of-information versus knock-out gap first observed on the LLaVA-1.5 screen (Appendix P).

Table 19: Equal-information controls under exact-576 (SCREEN, four-task macro accuracy %, first-token readout). *Best heuristic* is the strongest input-level selector in each column (Table 1); GRADFULL, IG8 and KNOCKOUT use the same 576-token reference prefill as CIG. Bold marks the best selector in each column.

Backbone	Selector	k=8	k=16	k=32	k=64	k=128	k=192
InternVL3-8B	Best heuristic	70.80	74.45	79.45	83.05	86.05	86.55
	KNOCKOUT	70.15	73.30	76.05	80.40	84.40	85.90
	IG8	73.25	76.60	79.40	82.90	85.05	85.85
	GRADFULL	74.95	80.15	83.80	85.65	**86.85**	87.00
	CIG	**81.80**	**83.55**	**85.40**	**86.25**	86.40	**87.30**
Qwen2.5-VL-7B	Best heuristic	68.75	72.90	76.75	81.30	83.05	84.20
	KNOCKOUT	71.05	76.05	79.65	82.05	85.05	85.40
	IG8	76.15	81.00	83.50	83.90	85.55	85.70
	GRADFULL	75.35	80.35	83.05	84.80	**85.85**	**86.10**
	CIG	**80.55**	**82.35**	**84.20**	**84.90**	85.30	85.75
InternVL3.5-4B	Best heuristic	65.75	71.70	73.90	77.25	80.00	82.20
	KNOCKOUT	66.10	68.75	72.20	75.60	79.80	81.30
	IG8	71.00	74.60	78.40	81.30	82.80	83.55
	GRADFULL	67.70	72.15	76.95	81.20	**83.45**	**83.60**
	CIG	**78.65**	**80.05**	**80.75**	**81.50**	82.50	82.45
Qwen3-VL-4B	Best heuristic	71.65	76.30	80.40	83.60	85.40	86.00
	KNOCKOUT	73.40	75.65	79.70	82.00	84.40	86.20
	IG8	77.50	82.10	85.15	86.45	**87.65**	**87.75**
	GRADFULL	75.85	81.05	84.55	85.60	87.35	87.70
	CIG	**82.25**	**84.15**	**86.05**	**86.55**	86.90	87.40

H PROTOCOL AND IMPLEMENTATION DETAILS

Environment. Single-GPU runs use one NVIDIA RTX 5090 (32 GiB); software versions are listed in the supplementary material.

Exact-576 backbones. Model identifiers and the input resolution chosen to realise $N = 576$ native visual tokens:

Backbone	Model id	Input px
InternVL3-8B	`OpenGVLab/InternVL3-8B`	672
Qwen2.5-VL-7B	`Qwen/Qwen2.5-VL-7B-Instruct`	672
InternVL3.5-4B	`OpenGVLab/InternVL3_5-4B`	672
Qwen3-VL-4B	`Qwen/Qwen3-VL-4B-Instruct`	768

Choices fixed before the main experiments. The definition of J_1; the DPC-$C = 8$ clustering configuration; the log-sum-exp merging of answer-option tokeniser variants; evaluation splits; and the seed.

Splits and macros. SCREEN is the first 500 examples of each task in dataset order; it is therefore a subset of FULL, and the exact-576 grid uses the same first-500 items on the native backbones. The held-out check on items 501–1,000 is in Appendix D. FULL splits are POPE 9,000; MMBench-EN-dev 4,329 (circular variants grouped by base question); SEED-Bench-image 14,233; ScienceQA-IMG 2,017; GQA testdev-balanced 12,578; MMMU-val 900; TextVQA-val 5,000; MME 2,374 — 50,431 items in total. MMBench-EN-dev circular variants are collapsed to base questions before any statistical test; a base question counts as correct only if all its variants are correct. MME enters no percentage macro. The seven-task macro weights tasks equally, not examples.

Budget semantics. Pre-LLM methods: k is the number of visual tokens entering the language model; VisionZip's merged contextual tokens count against k, although merged tokens are value substitutions rather than deletions. In-LLM methods: k is the layer-averaged number of retained visual tokens, obtained by bisection over the method's retention parameter and cached in a lookup table. KV-cache length is reported separately because in-LLM methods cannot shorten the prefix.

I BUDGET REACHABILITY FOR IN-LLM BASELINES

Three published methods cannot realise some requested budgets under their published schedules. We record these cells as NA rather than altering the published configuration.

Table 20: Reachability of requested budgets. NA cells are structural, not missing runs.

Method	Reachable	Unreachable	Reason
FastV	$k \geq 36$	$k \in \{8, 16, 32\}$	Official $K = 2$ schedule keeps all 576 tokens in layers 0–1: layer average ≥ 36.
SparseVLM	$k \in \{64, 96, 128, 192\}$	$k \in \{8, 16, 32\}$	Official `score.py` hard-codes retention schedules for those budgets only.
PyramidDrop	$k \geq 144$	$k \in \{16, 32, 64\}$	Official stages $[8, 16, 24]$ keep all 576 tokens for the first 8 of 32 layers.

FastV. The official implementation masks deep-layer tokens with an attention mask rather than deleting them, so accuracy and efficiency are measured on different code paths: the default implementation for accuracy and the in-place KV variant for latency. Under the official $K = 2$ configuration at $k = 64$, layers 0–1 retain all 576 visual tokens and the remaining 30 layers retain 30, for a layer average of 64.

SparseVLM. SparseVLM is run in a separate environment because its custom LLaMA modelling module would replace the attention path shared by all other methods; its official builder uses FP16, so these rows are labelled FP16 rather than BF16.

PyramidDrop. Under a unified layer-average budget the published stage locations imply a lower bound of 144 average retained tokens, so all 18 requested cells are structurally NA.

Baseline verification. All baselines are run from their official implementations with their published default settings and were not tuned. Since none of the repositories provides reference numbers for our exact protocol, we verified their selection rules, retained-token counts and end-to-end execution rather than reproducing published tables.

Screening table. Six-task SCREEN results for the baselines are collected in the table below.

Table 21: Six-task SCREEN (500 examples/task), LLaVA-1.5-7B, BF16, seed 42. Not comparable with FULL tables. SparseVLM runs in FP16 (see text).

Method	k	POPE	MMBench	SEED	SQA	GQA	MMMU	Macro
DivPrune	16	76.2	65.6	55.4	67.2	50.8	35.6	58.47
	32	81.4	70.2	60.2	67.2	52.4	35.0	61.07
	64	84.0	72.8	63.6	67.8	55.4	36.6	63.37
CDPruner	16	83.0	69.6	60.6	67.2	54.6	35.4	61.73
	32	85.0	73.0	64.6	70.2	56.6	36.4	64.30
	64	85.4	72.4	66.4	67.2	56.8	37.8	64.33
VisionZip	16	67.4	63.2	45.6	66.6	44.2	34.0	53.50
	32	74.0	68.6	55.0	68.2	49.6	35.0	58.40
	64	78.6	71.2	60.0	66.8	51.4	36.2	60.70
FastV	64	macro 50.97; $k < 36$ unreachable						
SparseVLM (FP16)	64	80.8	73.2	60.2	68.4	52.0	36.2	61.80
PyramidDrop	—	all requested budgets structurally NA						
MMTok	32	82.4	71.2	60.8	68.2	56.0	35.6	62.37
Random	32	72.6	67.2	57.2	66.4	49.0	32.8	57.53

J LLAVA-1.5-7B FULL SPLITS AND PAIRED STATISTICS

Per-sample paired comparisons on FULL splits at $k = 32$ (94,532 paired units in total). Confidence intervals are 10,000-draw stratified paired bootstrap; the macro resamples within task then weights

tasks equally. TextVQA keeps fractional native VQA credit for scores and bootstrap; its exact McNemar test binarises correctness at native score $= 1.0$. MME preserves official image-pair/category aggregation, so its item-level McNemar value is supporting evidence only.

Table 22: Paired comparisons at $k = 32$ on FULL splits. The attribution analysis of Appendix R uses a separate evaluation batch whose accuracy levels differ from those here by up to 0.9 pp (run-environment variation); it is used only for category attribution.

Comparison	Task	Left	Right	Δ	95% CI	McNemar p
CIG vs MMTok	POPE	85.578	84.989	$+0.589$	$[+0.011, +1.167]$	5.1×10^{-2}
	MMBench	60.739	59.021	$+1.718$	$[+0.000, +3.436]$	6.7×10^{-2}
	SEED	64.428	59.777	$+4.651$	$[+4.005, +5.291]$	9.0×10^{-46}
	ScienceQA	69.658	68.964	$+0.694$	$[-0.694, +2.033]$	3.5×10^{-1}
	GQA	59.238	56.472	$+2.767$	$[+2.107, +3.443]$	5.9×10^{-16}
	MMMU	35.000	34.889	$+0.111$	$[-1.889, +2.111]$	1.0
	TextVQA	32.914	37.414	-4.500	$[-5.604, -3.410]$	5.9×10^{-14}
	MME (raw)	1795.9	1678.0	$+117.98$	$[+46.32, +189.25]$	4.8×10^{-7}
	Macro$_7$	58.222	57.361	$\mathbf{+0.861}$	$[+0.384, +1.328]$	1.9×10^{-31}
CIG vs CDPruner	POPE	85.578	87.522	-1.944	$[-2.556, -1.333]$	6.1×10^{-10}
	MMBench	60.739	60.739	$+0.000$	$[-1.632, +1.632]$	1.0
	SEED	64.428	60.915	$+3.513$	$[+2.895, +4.131]$	4.3×10^{-28}
	ScienceQA	69.658	68.617	$+1.041$	$[-0.297, +2.429]$	1.7×10^{-1}
	GQA	59.238	57.354	$+1.884$	$[+1.232, +2.536]$	1.7×10^{-8}
	MMMU	35.000	35.000	$+0.000$	$[-1.889, +1.889]$	1.0
	TextVQA	32.914	36.264	-3.350	$[-4.428, -2.254]$	6.4×10^{-10}
	MME (raw)	1795.9	1674.9	$+121.02$	$[+45.61, +198.04]$	9.1×10^{-8}
	Macro$_7$	58.222	58.059	$\mathbf{+0.163}$	$[\mathbf{-0.305}, \mathbf{+0.636}]$	5.3×10^{-8}

The macro interval against CDPruner contains zero, so the two methods are indistinguishable at $k = 32$ in aggregate. The pooled item-level McNemar value tests a different null: it pools items across tasks with unequal sizes and is therefore driven by the large SEED-Bench and GQA splits. The per-task intervals, which do separate on four of seven tasks in opposite directions, are the more informative summary.

K BUDGET CURVES AND TREND TABLES

Table 23: SCREEN budget curve (four tasks $\times$ 500, macro %). NA cells are unreachable under the method's published schedule; J-oracle uses labels and is not deployable.

Method	k=8	k=16	k=32	k=64	k=128	k=192
Random	56.80	61.20	65.85	68.60	72.00	72.25
FastV	NA	NA	NA	57.75	66.65	68.80
VisionZip	53.85	60.70	66.45	69.15	70.95	72.40
DivPrune	59.80	66.10	69.75	72.05	72.55	73.50
MMTok	66.50	69.45	70.65	71.70	72.35	72.85
CDPruner	67.45	70.10	73.20	72.85	73.60	73.70
CIG	**71.45**	**72.35**	**73.60**	**74.00**	**73.90**	**74.35**
J-oracle	82.45	83.65	85.55	86.40	86.20	85.45

Two observations. First, CIG's per-task profile is stable across budgets: it leads on SEED-Bench, GQA, MMBench and ScienceQA at all three budgets and trails on TextVQA at all three. Second, the aggregate crossover between $k = 16$ and $k = 64$ is driven almost entirely by TextVQA: excluding it, CIG leads CDPruner by $+1.86$ pp, $+0.75$ pp and $+0.58$ pp at $k = 16/32/64$ respectively. The reported macros include TextVQA.

Table 24: FULL per-task trend rows at $k=16$ and $k=64$ (eight tasks; MME raw excluded from the macro). The $k=32$ rows are Table 4.

Method	k	POPE	MMBench	SEED	SQA	GQA	MMMU	TextVQA	MME	AVG$_7$
VisionZip	16	66.96	47.08	46.95	67.63	46.65	34.00	31.07	1352.7	48.62
MMTok	16	82.87	54.55	57.21	69.16	54.36	**35.44**	30.97	1585.0	54.94
CDPruner	16	**87.32**	56.96	59.24	68.42	55.26	34.78	29.82	1609.6	55.97
CIG	16	85.47	**58.51**	**63.23**	**69.61**	**58.30**	34.89	28.93	**1761.2**	**56.99**
VisionZip	64	80.59	60.14	57.61	68.86	55.23	**36.00**	**42.04**	1708.5	57.21
MMTok	64	86.69	61.43	61.49	68.96	58.28	35.00	41.49	1711.6	59.05
CDPruner	64	**87.67**	61.51	62.62	67.97	58.84	36.11	40.73	1715.7	**59.35**
CIG	64	86.39	**62.71**	**65.05**	**69.61**	**60.19**	35.00	36.46	**1805.8**	59.34

L LLaVA-NeXT (HIGH RESOLUTION)

Candidate pools are the concatenated global AnyRes patch tokens, 1,728–2,880 per sample. VisionZip and CDPruner use model-native AnyRes global adaptations: crop-local signals and features are concatenated and the method is invoked once at the total budget.

Table 25: LLaVA-NeXT-7B, $k=128$, four FULL splits (POPE 9,000; GQA 12,578; TextVQA 5,000; MMBench 4,329).

Method	POPE	GQA	TextVQA	MMBench	Macro
Random	75.13	55.77	24.09	56.10	52.77
VisionZip	77.94	54.52	**51.28**	61.17	61.23
CIG	86.16	60.18	40.36	63.66	62.59
MMTok	84.51	60.38	48.08	**64.26**	64.31
CDPruner	**87.97**	**60.45**	50.13	64.18	**65.68**
FastV	NA	NA	NA	NA	NA

Table 26: LLaVA-NeXT-7B SCREEN sweep (four tasks × 500). FastV cannot reach the two smaller budgets.

Method	k	POPE	GQA	TextVQA	MMBench	Macro
Random	64	71.20	50.80	17.62	69.40	52.26
	128	75.40	53.00	24.88	70.40	55.92
	320	81.60	59.00	35.20	72.40	62.05
FastV	64	NA	NA	NA	NA	NA
	128	NA	NA	NA	NA	NA
	320	70.20	44.20	41.32	64.40	55.03
VisionZip	64	71.80	49.80	38.46	65.20	56.31
	128	79.60	54.80	49.78	70.60	63.70
	320	86.40	58.80	56.16	72.80	68.54
CDPruner	64	87.00	60.20	40.56	73.60	**65.34**
	128	86.80	61.80	46.18	74.40	67.30
	320	87.40	62.20	51.84	74.20	68.91
MMTok	64	83.80	55.20	40.64	72.40	63.01
	128	86.40	61.60	47.56	74.00	67.39
	320	87.40	63.60	52.88	74.20	**69.52**
CIG	64	87.00	60.80	37.42	73.60	64.70
	128	87.20	60.40	42.24	74.60	66.11
	320	87.20	61.80	46.16	74.60	67.44

Cluster-count selection. C is the only hyper-parameter re-selected for this backbone, by the highest four-task SCREEN macro at $k=64$ (ties to smaller C), and was fixed before the FULL runs. Results: $C=8 \to 64.705$, $C=16 \to 61.215$, $C=24 \to 61.230$. That *more* clusters hurt is informative: with a fixed budget k, increasing C shrinks the per-cluster token supply, so the round-robin allocator is forced to draw from many clusters and the greedy search loses the ability to concentrate the budget. A granularity scheme for large candidate pools therefore has to change the allocator, not just C.

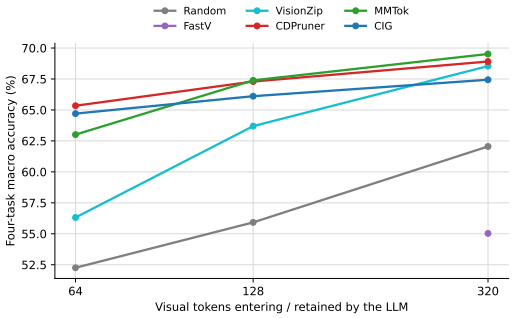


Figure 3: LLaVA-NeXT-7B SCREEN macro by budget. CIG is second at $k = 64$; the AnyRes-adapted CDPruner and MMTok lead at larger budgets.

M READOUT VALIDATION

Readout. All native results in the exact-576 grid use an answer-option first-token readout with true visual-token deletion. The InternVL models use contiguous post-deletion positions; the Qwen models retain native M-RoPE positions with the position grid pruned synchronously with the tokens. When all tokens are retained, per-logit agreement with the unmodified model is exact.

Readout versus generation. Table 27 compares the first-token readout with free-form greedy generation followed by the `lmms-eval` task parser, on 200 SCREEN items per task for the unpruned model and for five pruned conditions. Parse failures count as disagreements. Agreement is at least 95% in 90 of the 96 backbone–task–condition cells; the six exceptions are all POPE on InternVL3.5-4B, where the unpruned model itself agrees on only 69.5% of items because it often answers with a sentence the parser does not map to yes/no, so the discrepancy is a property of that backbone's POPE generations rather than of any selector (its other three tasks agree on at least 98%). Under generation scoring, CIG@32 remains ahead of MMTok@32 and CDPruner@32 on every backbone.

Table 27: Readout-versus-generation agreement on the four multiple-choice tasks (SCREEN, exact-576, $n = 200$ per task and condition). For each condition: mean and minimum agreement over the four tasks between the first-token readout and free-form greedy generation with the `lmms-eval` parser, and four-task macro accuracy under generation scoring. The only conditions below the 95% threshold are POPE on InternVL3.5-4B, for every method including the unpruned model.

	InternVL3-8B			Qwen2.5-VL-7B			InternVL3.5-4B			Qwen3-VL-4B		
Condition	Agr.	min	Acc.	Agr.	min	Acc.	Agr.	min	Acc.	Agr.	min	Acc.
Full@576	0.989	0.980	87.87	0.994	0.990	84.38	0.921	0.695	80.12	0.989	0.975	87.50
CIG@32	0.974	0.970	84.38	0.991	0.985	81.88	0.951	0.815	80.75	0.991	0.980	85.38
CIG@64	0.978	0.960	85.75	0.996	0.995	83.25	0.912	0.670	77.25	0.992	0.985	86.38
MMTok@32	0.972	0.950	81.88	0.988	0.980	76.62	0.963	0.875	76.50	0.988	0.975	81.00
CDPruner@32	0.980	0.965	79.62	0.989	0.980	73.38	0.914	0.680	71.12	0.995	0.990	75.62
Random@32	0.965	0.955	80.25	0.979	0.955	76.75	0.921	0.700	73.25	0.990	0.980	76.25

In-LLM methods are excluded from every native track for a protocol reason: their budgets are defined per layer inside the language model, so they cannot be put on the same "tokens entering the model" axis as pre-LLM selection under true deletion (Appendix F).

N FIXED-KV WORKLOAD: FULL GRIDS

Protocol: exact-576, InternVL3-8B (672 px) and Qwen3-VL-4B (768 px), $k = 32$, generation length $L \in \{128, 256, 512, 1024\}$, batch $\in \{4, 16\}$, 20 warm-up + 200 measured requests per cell, RTX 5090 32 GiB, seed 42, BF16. Every compressed row's *wall* column includes that method's selection cost; *deploy* excludes it. Peak is process-level `max_memory_allocated`, not a KV-only delta. Accuracy is taken from the exact-576 grid and is not remeasured here.

Reading the grids. The crossover is conditional on memory pressure, not on batch size alone. InternVL3-8B reaches 18.55 GiB uncompressed at batch 16 and crosses at every length; Qwen3-VL-

Table 28: Full batch grids. Δ is the reduction in wall $p50$ against Full at the same backbone, length and batch. Positive is faster.

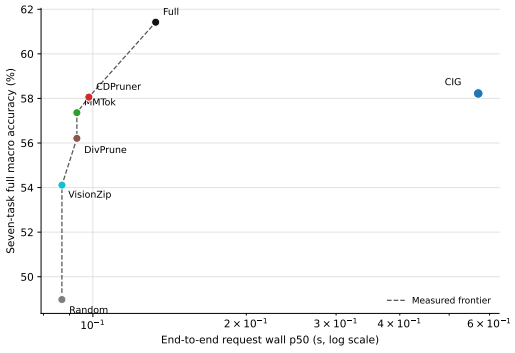
Backbone	L	Batch	Method	Wall $p50$	Deploy $p50$	Sel. (s)	Δ	tok/s	Peak (GiB)
InternVL3-8B	128	4	Full	2.894	2.894	0.000	—	193	15.80
			MMTok	2.605	2.580	0.025	**+10.0%**	197	15.02
			CIG	3.249	2.745	0.504	-12.3%	193	15.02
			Batched CIG	2.956	2.725	0.231	-2.1%	193	15.02
		16	Full	3.688	3.688	0.000	—	734	18.55
			MMTok	2.830	2.804	0.025	**+23.3%**	775	15.46
			CIG	3.270	2.766	0.504	**+11.3%**	784	15.46
			Batched CIG	2.853	2.622	0.231	**+22.6%**	810	15.46
	256	4	Full	5.502	5.502	0.000	—	192	15.80
			MMTok	5.471	5.446	0.025	**+0.6%**	190	15.02
			CIG	6.016	5.511	0.504	-9.3%	189	15.02
			Batched CIG	5.699	5.469	0.231	-3.6%	191	15.02
		16	Full	6.646	6.646	0.000	—	712	18.55
			MMTok	5.531	5.506	0.025	**+16.8%**	766	15.46
			CIG	6.077	5.573	0.504	**+8.6%**	756	15.46
			Batched CIG	5.719	5.489	0.231	**+13.9%**	767	15.46
	512	4	Full	10.893	10.893	0.000	—	192	15.80
			MMTok	10.104	10.079	0.025	**+7.2%**	201	15.05
			CIG	10.729	10.225	0.504	**+1.5%**	198	15.05
			Batched CIG	10.408	10.177	0.231	**+4.5%**	198	15.05
		16	Full	13.045	13.045	0.000	—	674	18.55
			MMTok	10.788	10.763	0.025	**+17.3%**	781	15.55
			CIG	10.707	10.202	0.504	**+17.9%**	808	15.55
			Batched CIG	10.963	10.732	0.231	**+16.0%**	782	15.55
	1024	4	Full	21.221	21.221	0.000	—	196	15.80
			MMTok	20.620	20.594	0.025	**+2.8%**	197	15.18
			CIG	21.716	21.211	0.504	-2.3%	194	15.18
			Batched CIG	21.559	21.329	0.231	-1.6%	193	15.18
		16	Full	27.847	27.847	0.000	—	608	18.55
			MMTok	22.425	22.399	0.025	**+19.5%**	733	16.10
			CIG	23.119	22.615	0.504	**+17.0%**	729	16.10
			Batched CIG	22.745	22.514	0.231	**+18.3%**	731	16.10
Qwen3-VL-4B	128	4	Full	3.571	3.571	0.000	—	146	8.92
			MMTok	3.437	3.400	0.036	**+3.8%**	150	8.48
			CIG	3.795	3.531	0.264	-6.3%	146	8.48
			Batched CIG	3.725	3.669	0.056	-4.3%	142	8.48
		16	Full	3.739	3.739	0.000	—	621	10.56
			MMTok	3.627	3.591	0.036	**+3.0%**	581	8.80
			CIG	3.848	3.585	0.264	-2.9%	589	8.80
			Batched CIG	3.638	3.582	0.056	**+2.7%**	595	8.80
	256	4	Full	6.922	6.922	0.000	—	148	8.92
			MMTok	7.305	7.269	0.036	-5.5%	141	8.55
			CIG	7.027	6.764	0.264	-1.5%	150	8.55
			Batched CIG	6.895	6.840	0.056	**+0.4%**	148	8.55
		16	Full	7.476	7.476	0.000	—	586	10.56
			MMTok	6.832	6.796	0.036	**+8.6%**	599	9.08
			CIG	7.278	7.014	0.264	**+2.7%**	593	9.08
			Batched CIG	7.043	6.987	0.056	**+5.8%**	598	9.08
	512	4	Full	14.264	14.264	0.000	—	147	9.01
			MMTok	13.863	13.827	0.036	**+2.8%**	147	8.69
			CIG	14.813	14.550	0.264	-3.9%	144	8.69
			Batched CIG	13.437	13.382	0.056	**+5.8%**	150	8.69
		16	Full	14.237	14.237	0.000	—	599	10.92
			MMTok	13.516	13.480	0.036	**+5.1%**	602	9.65
			CIG	14.418	14.155	0.264	-1.3%	583	9.65
			Batched CIG	14.223	14.167	0.056	**+0.1%**	587	9.65
	1024	4	Full	27.255	27.255	0.000	—	148	9.29
			MMTok	29.220	29.184	0.036	-7.2%	142	8.97
			CIG	29.255	28.992	0.264	-7.3%	144	8.97
			Batched CIG	27.430	27.374	0.056	-0.6%	147	8.97
		16	Full	26.859	26.859	0.000	—	605	12.06
			MMTok	27.440	27.404	0.036	-2.2%	597	10.79
			CIG	28.535	28.272	0.264	-6.2%	596	10.79
			Batched CIG	28.934	28.878	0.056	-7.7%	574	10.79

4B peaks at 10.6–12.1 GiB and does not. At batch 4 neither backbone crosses reliably (CIG is faster than Full in 1 of 8 cells), and at batch 1 the earlier LLaVA-1.5 conclusion stands. Selection is a fixed per-request cost (0.504 s on InternVL3-8B, 0.264 s on Qwen3-VL-4B; batched: 0.231 and 0.056 s, i.e. batching makes selection $2.2\times$ and $4.7\times$ cheaper at $k = 32$ in this serving pipeline, within the

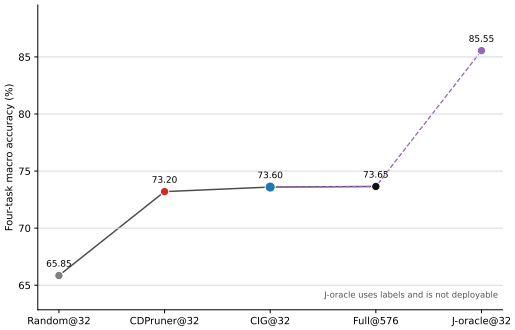
per-backbone ranges of Table 3), so its share of wall time falls as L grows—from 15.4% at $L=128$ to 2.2% at $L=1024$ on InternVL3-8B—which is the mechanism behind the crossover.

Multi-image prefill. Repeating exact-576 visual blocks $r \in \{1, 4, 8, 16\}$ times leaves CIG's deployment prefill flat ($0.057 \rightarrow 0.061$ s) while the uncompressed prefill grows $9.5\times$ ($0.061 \rightarrow 0.582$ s). The measurement uses $L=1$ on one backbone and excludes selection, which would itself scale with r in a multi-image request; it isolates the deployment-side benefit of a fixed visual budget. All timings in this appendix use the eager attention path.

O EFFICIENCY DETAILS



(a) Accuracy versus measured latency at batch 1. CIG trades single-request wall-clock for accuracy at a fixed small KV budget.



(b) Headroom ladder at $k=32$. The label-using oracle over the same clusters is 11.95 pp above CIG.

Figure 4: (a) the single-request cost of intervention; (b) the headroom that remains at $k=32$.

Table 29: **Measured efficiency** on LLaVA-1.5-7B (POPE, 20 warm-up + 200 measured requests, single GPU, batch 1). Accuracy is the seven-task FULL macro from Table 4. *Deploy TF* counts only the visual tokens the language model consumes; *Select TF* is the additional selection compute. Peak memory is the process-level CUDA maximum, not a KV-only delta. *Wall* aggregates the measured pipeline stages (image encoding, selection, prefill and decode; Table 30); the end-to-end request wall-clock in Table 30 additionally includes tokenisation and request handling and is therefore larger.

Method	k	AVG$_7$	Wall p50 (s)	Wall p90 (s)	Peak (GiB)	Deploy TF	Select TF
Full	576	61.42	0.1328	0.1511	14.001	8.064	—
Random	32	48.98	0.0869	0.1086	13.691	0.448	≈ 0
VisionZip	32	54.11	0.0870	0.0928	13.954	0.448	≈ 0
DivPrune	32	56.21	0.0930	0.1115	13.685	0.448	≈ 0
MMTok	32	57.36	0.0930	0.1096	13.685	0.448	≈ 0
CDPruner	32	58.06	0.0982	0.1156	13.685	0.448	≈ 0
CIG	32	**58.22**	0.5701	0.7116	13.695	0.448	13.614

CIG has the fastest prefill (0.0213 s versus 0.0695 s for the uncompressed model) and decode (0.0230 s versus 0.0474 s) of any configuration, because it admits the fewest tokens; its entire disadvantage is in the pre-LLM stage, which grows with the budget (0.4455/0.5340/0.6635 s at $k=16/32/64$) because larger budgets make each probe prefill longer.

FLOPs accounting. With the estimator of 0.014 TF per visual token, CIG@32 costs 13.614 TF of selection plus 0.448 TF of deployment, i.e. 14.062 TF. Matching this with MMTok would need $k'=1004$ visual tokens, above LLaVA's 576 maximum, so no equal-FLOPs MMTok row exists on this backbone; MMTok@576 (8.064 TF) is the closest runnable upper bound. The largest single term in CIG's selection cost is the 576-token reference prefill (8.064 TF), the price of knowing $p_\star$; the probes themselves average 401 tokens.

At 128 generated tokens the decode phase is essentially budget-independent on this backbone, so CIG's 0.5876 s selection cost is not amortised: it is $+0.669$ s against MMTok and $+0.692$ s against

Table 30: Generation-phase profile (POPE, first 220 requests: 20 warm-up, 200 measured; batch 1, seed 42, BF16). *Pre-LLM* time is image encoding plus selection for pre-LLM methods; FastV reports image encoding only because its pruning runs inside the language model. Peak allocation is the process-level CUDA maximum. *Wall* is the end-to-end request wall-clock and exceeds the sum of the stage medians below by tokenisation and request-handling overhead; Table 29 reports the stage-aggregate wall time.

Method	k	Wall p50 / mean	Prefill p50 / mean	Decode p50 / mean	Peak (GiB)	Pre-LLM p50 / mean
Full	576	0.1888 / 0.1925	0.0695 / 0.0702	0.0474 / 0.0538	14.00	0.0092 / 0.0097
Random	32	0.1402 / 0.1480	0.0223 / 0.0288	0.0245 / 0.0317	13.70	0.0126 / 0.0145
MMTok	32	0.1533 / 0.1619	0.0227 / 0.0299	0.0522 / 0.0614	13.69	0.0126 / 0.0155
DivPrune	32	0.1472 / 0.1595	0.0232 / 0.0311	0.0534 / 0.0589	13.69	0.0125 / 0.0127
CDPruner	32	0.1599 / 0.1681	0.0223 / 0.0293	0.0628 / 0.0697	13.69	0.0211 / 0.0212
VisionZip	32	0.1482 / 0.1568	0.0218 / 0.0277	0.0570 / 0.0628	13.95	0.0194 / 0.0195
FastV	64	0.1200 / 0.1236	0.0239 / 0.0281	0.0407 / 0.0429	13.79	0.0090 / 0.0092
CIG	16	0.5698 / 0.5902	0.0218 / 0.0236	0.0234 / 0.0253	13.70	0.4455 / 0.4712
CIG	32	0.6519 / 0.6546	0.0213 / 0.0232	0.0230 / 0.0251	13.70	0.5340 / 0.5399
CIG	64	0.7672 / 0.7737	0.0216 / 0.0232	0.0234 / 0.0250	13.72	0.6635 / 0.6613

Table 31: Long-generation amortisation. 200 unique COCO-val images (Chen et al., 2015), one fixed detailed-description prompt, 128 generated tokens per request. Latency only.

Method	n	vis. tok.	Wall p50 / mean (s)	Decode mean (s)	tok/s mean	Select mean (s)
Full	200	576	2.6095 / 2.6182	2.5279	50.76	—
MMTok	200	32	2.6678 / 2.6409	2.5884	49.54	—
CIG	200	32	3.2368 / 3.3103	2.6580	48.50	0.5876

the uncompressed model. At this operating point the efficiency case for CIG rests on KV memory, not on single-stream wall-clock.

P OBJECTIVE AND INTERVENTION FORM (LLAVA-1.5 SCREEN)

This table is on the LLaVA-1.5 four-task SCREEN protocol and is retained for the axis the exact-576 ablation does not cover, the knock-out intervention form. For the objective axis, the per-cell values on native backbones are in Appendix T, where J_2 and J_3 cost 9.44 and 4.17 pp; the two protocols are reported separately.

Table 32: Configuration choices, four-task SCREEN macro at $k=32$. All settings were fixed on this screen before the main experiments.

Axis	Setting	Macro
Objective	$J_1 = -\mathrm{KL}(p_\star \| p_S)$ **(selected)**	**73.60**
	J_2/J_3 (confidence gain / mixed form)	71.15
	$J_{\text{oracle}} = p_S(y^\star)$ (uses labels; upper bound)	85.55
Intervention form	Gain of information, real subset prefill **(selected)**	**73.60**
	Knock-out (delete one cluster from full)	64.95

Two observations. First, the gap between gain-of-information and knock-out scoring (73.60 versus 64.95) exceeds the gap between CIG and any baseline, although both are intervention-based: at $k=32$ the deployed condition is near-empty, so "what does adding this cluster to an empty context buy?" is far more informative than "what does removing it from a full context cost?". Second, the label-using oracle at 85.55 is the number shown in Figure 4b; only the objective differs, so the 11.95 pp gap is attributable to the objective and the greedy search rather than to the candidate pool.

Seed stability. Re-running CIG and MMTok with seeds 42, 43 and 44 on the four-task SCREEN at $k=32$ gives identical results across seeds. This reflects determinism rather than robustness: given a fixed image and prompt both selectors are deterministic, and the seed enters only through tie-breaking in clustering, which these images do not trigger. Variance under data resampling is captured by the bootstrap intervals in Appendix J.

Q SIGNAL CORRELATIONS AND COMPLEMENTARITY

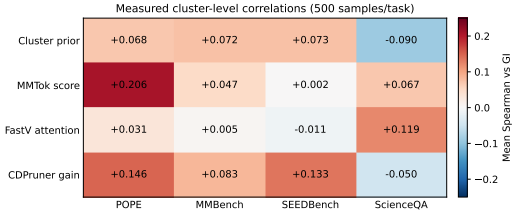


Figure 5: Per-task Spearman correlation between the first-step marginal J_1 gain and four heuristic signals, computed on the same fixed clusters (four tasks $\times$ 500 examples).

Table 33: Left: correlation between the intervention signal and heuristic scores. Right: pairwise overlap of the token sets selected at $k=32$ (same items, $n=2000$).

Signal vs J_1	Spearman	Selection pair	Jaccard	Union size
MMTok coverage marginal	0.0805	CIG / MMTok	0.0748	59.62
CDPruner kernel gain	0.0781	CIG / CDPruner	0.0726	59.82
FastV layer-2 attention	0.0362	MMTok / CDPruner	0.0790	59.42
Cross-modal prior	0.0307			

Per-task Spearman values against CDPruner's kernel gain are 0.146 (POPE), 0.083 (MMBench), 0.133 (SEED) and -0.050 (ScienceQA); the largest single value observed against any heuristic is 0.206 (MMTok on POPE). A two-stage design in which a heuristic proposes candidates and J_1 re-ranks them would be attractive only if some heuristic tracked the intervention gain closely (we set $\rho_s \geq 0.6$ as the bar); none approaches it.

Re-ranking within the union reaches 72.90, *below* CIG alone (73.60), so we do not propose a union variant. The oracle within the same union reaches 83.90, confirming that the complementarity is real but that our objective cannot exploit it.

Table 34: Constrained union experiment. Candidate pool is CIG@32 ∪ CDPruner@32 (mean 59.8 tokens); both rows select 32 tokens from that pool.

Method	POPE	MMBench	SEED	ScienceQA	Macro
J_1 re-rank within union	84.20	74.20	65.80	67.40	72.90
Label oracle within union	94.40	82.60	79.60	79.00	83.90
CIG alone (reference)	85.40	74.20	66.60	68.20	73.60

Both union rows use an identical candidate pool and budget and differ only in the objective (J_1 versus the label), so their 11.00 pp separation is attributable to objective mismatch alone. Since the full CIG-to-oracle gap over cluster-derived subsets is 11.95 pp (Figure 4b), at most ≈1 pp of that gap is due to the candidate space and the greedy search; the remainder reflects the fact that maximising agreement with $p_\star$ is not the same as maximising label accuracy, most visibly when the full model is itself wrong. The oracle gap should therefore not be read as search headroom.

R GAIN ATTRIBUTION

Evaluation batch. The attribution tables in this appendix come from a separate evaluation batch than the one behind Tables 4 and 22. BF16 inference is not bitwise reproducible across such changes, so individual item outcomes differ between the two batches and per-task accuracy levels differ by up to 0.9 pp (e.g. CIG on POPE: 86.47 here versus 85.58 in Table 4). We therefore use this batch only for attribution.

Table 35: **Where the gains come from.** Accuracy (%) by question type on 26,414 paired FULL units at $k = 32$ on LLaVA-1.5-7B, categories assigned deterministically from official dataset labels (CIG versus MMTok; separate evaluation batch, see above). The per-benchmark decomposition and the per-category MMBench breakdown, including categories where CIG loses, follow below.

Question type	n	CIG	MMTok	Δ
Counting	2,447	56.89	50.39	+6.50
Fine-grained attribute	5,067	67.40	61.08	+6.32
Spatial relation	1,973	51.65	46.93	+4.71
OCR / text	202	35.64	33.17	+2.48
Object hallucination	14,183	80.87	78.76	+2.12
Knowledge	2,542	69.20	68.96	+0.24

Table 36: Per-benchmark decomposition of the six question types. *Net* is net correct answers gained; *only* columns count samples solved by exactly one method.

Benchmark	Category	n	CIG	MMTok	Δ	Net	CIG-only / MMTok-only
POPE	object hallucination	9,000	86.47	85.08	+1.39	+125	407 / 282
MMBench	object hallucination	194	87.11	87.11	0.00	0	7 / 7
MMBench	spatial relation	241	37.34	32.78	+4.56	+11	19 / 8
MMBench	OCR / text	117	35.04	37.61	−2.56	−3	1 / 4
MMBench	fine-grained attr.	418	72.01	66.27	+5.74	+24	36 / 12
MMBench	knowledge	194	58.25	61.86	−3.61	−7	3 / 10
SEED	object hallucination	4,989	70.54	67.03	+3.51	+175	400 / 225
SEED	counting	2,447	56.89	50.39	+6.50	+159	306 / 147
SEED	spatial relation	1,732	53.64	48.90	+4.73	+82	167 / 85
SEED	OCR / text	85	36.47	27.06	+9.41	+8	14 / 6
SEED	fine-grained attr.	4,649	66.98	60.62	+6.37	+296	541 / 245
SEED	knowledge	331	74.62	72.51	+2.11	+7	19 / 12
ScienceQA	knowledge	2,017	69.36	69.06	+0.30	+6	104 / 98

Categories are assigned deterministically from official dataset labels. MMBench circular variants are collapsed to base questions first. Note that MMBench's OCR and knowledge rows go the other way from SEED's, and that the OCR category has only 202 pooled samples, so the aggregate OCR figure in Table 35 is weakly supported; the TextVQA result in Table 4, where CIG trails, is the more reliable OCR evidence.

Note the tension between the aggregate spatial-relation gain in Table 35 (+4.71 pp, driven by SEED-Bench) and MMBench's `spatial_relationship` category (−4.44 pp on 45 base questions).

Table 37: MMBench-EN-dev per-category attribution (base questions, circular variants collapsed). Both directions are shown; categories where CIG loses are in the lower block.

Category	n	CIG	MMTok	Δ	Category	n	CIG	MMTok	Δ
image quality	53	47.17	20.75	+26.42	image scene	104	94.23	93.27	+0.96
attribute comparison	44	52.27	40.91	+11.36	celebrity recog.	99	74.75	74.75	0.00
object localization	81	35.80	27.16	+8.64					
image style	53	75.47	67.92	+7.55	CIG *loses:*				
social relation	43	60.47	53.49	+6.98	image emotion	50	78.00	80.00	−2.00
nature relation	48	39.58	35.42	+4.17	physical prop. reas.	75	48.00	50.67	−2.67
physical relation	24	37.50	33.33	+4.17	action recognition	54	77.78	81.48	−3.70
image topic	36	80.56	77.78	+2.78	spatial relationship	45	15.56	20.00	−4.44
OCR	39	58.97	56.41	+2.56	struct. image-text	78	23.08	28.21	−5.13
future prediction	40	47.50	45.00	+2.50	function reasoning	79	73.42	81.01	−7.59
identity reasoning	45	95.56	93.33	+2.22					
attribute recognition	74	77.03	75.68	+1.35					

Per-category MMBench cells are small and circular-variant collapsing makes them stricter than raw accuracy, so we treat the pooled cross-task table as the primary attribution evidence and this table as a finer-grained but noisier view.

S SELECTION GEOMETRY

Normalised 6×6 spatial entropy is 0.676 for CIG against 0.789 for MMTok and 0.756 for CDPruner, and CIG places more mass in the outermost patch band (0.290 versus 0.256 and 0.239).

Table 38: Spatial statistics of selected tokens on the 24×24 patch grid (four tasks $\times$ 500, $k=32$). *Edge* is the outermost 3-patch band; entropy is normalised over 6×6 spatial bins.

Method	Centre dist.	Edge frac.	Entropy	Quadrants	Centroid (row, col)
CIG	0.4858	0.2896	0.6760	3.766	(0.512, 0.490)
MMTok	0.4720	0.2558	0.7886	3.999	(0.506, 0.487)
CDPruner	0.4506	0.2392	0.7556	3.996	(0.508, 0.504)

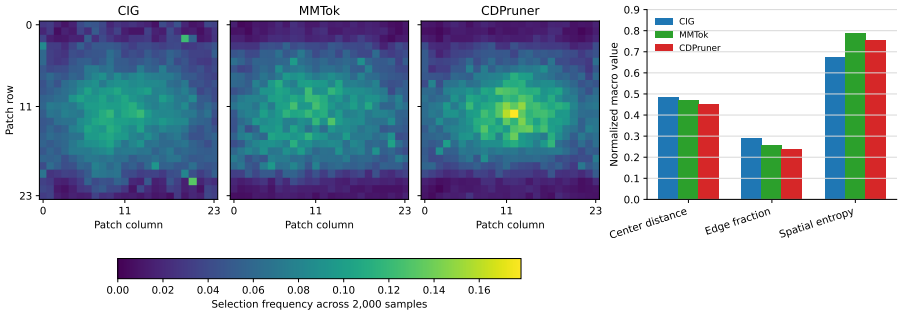


Figure 6: 24×24 selection-frequency maps and macro statistics per method. All three are centred (centroids within 0.02 of the grid centre); the methods differ in concentration, not in central bias.

Per task, CIG's edge fraction ranges from 0.249 (SEED-Bench) to 0.348 (ScienceQA-IMG), the largest spread of the three methods, consistent with a content-dependent rather than a fixed spatial policy. None of the three shows the end-of-sequence bias reported for attention-based scores (Wen et al., 2025a).

T CANDIDATE-GENERATION ABLATION (FULL)

Protocol: exact-576, InternVL3-8B and Qwen3-VL-4B, four tasks $\times$ 500 samples, $k\in\{8,32,128\}$, seed 42, BF16, batch 1; one component changed per row against the reference configuration, which is that of Table 1.

Table 39: **Candidate-generation ablation, exact-576.** Each row changes one component of the reference configuration (C=8, J_1, M=8, round-robin, cross-modal priors, seed 42) and reports the four-task macro change against it, on two backbones at three budgets (4×500 examples per cell). *Probes* is the mean number of subset prefills per request, averaged over all six cells. Rows are ordered by |mean Δ|.

		InternVL3-8B			Qwen3-VL-4B				
Component	Variant	k=8	k=32	k=128	k=8	k=32	k=128	mean	probes
Objective	J_2 confidence gain	−11.30	−10.50	−5.40	−10.30	−11.30	−7.85	−9.44	14.8
Objective	J_3 mixed	−2.75	−4.15	−2.45	−3.25	−6.10	−6.30	−4.17	14.2
Fan-out	C=16, M=4	−8.15	−8.50	−4.45	−4.60	−5.15	−3.05	−5.65	10.9
Fan-out	C=16, M=8	−1.90	−1.65	−0.15	−0.40	−1.45	+0.10	−0.91	23.4
Fan-out	C=16, M=16	+2.75	+1.15	+0.55	+2.05	+0.70	+0.40	+1.27	46.8
Cluster count	C=4, M=8	−3.20	−2.60	−1.05	−2.35	−2.20	−0.45	−1.97	5.1
Cluster count	C=16, M=8	−1.90	−1.65	−0.15	−0.40	−1.45	+0.10	−0.91	23.4
Cluster count	C=32, M=8	−3.90	−4.55	−1.85	−0.70	−1.80	−0.65	−2.24	39.1
Search depth	C=32, $\leq$8 tok./round	−2.40	−3.75	−1.35	+0.05	−1.15	−0.10	−1.45	54.8
Allocator	sequential fill	−0.40	−1.25	−0.60	−3.65	−2.50	+0.05	−1.39	14.7
Allocator	proportional	−0.20	−0.15	−0.40	+0.00	−0.05	+0.30	−0.08	14.6
Clustering	k-means	+0.70	+0.25	+0.85	+0.35	+0.05	+0.80	+0.50	11.8
Clustering	agglomerative	−0.35	−0.05	+1.05	+0.50	+0.05	+0.60	+0.30	12.2
Member prior	uniform spatial	+0.25	+0.30	+0.00	−1.75	−0.85	+0.35	−0.28	14.5
Member prior	random within cluster	+0.25	+0.40	+0.10	−1.10	−0.45	+0.45	−0.06	14.5
Clustering	+ spatial link (γ=0.25)	+0.10	+0.15	−0.35	+0.20	−0.25	−0.10	−0.04	14.5
Cluster prior	cluster size	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	14.2
Cluster prior	random	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	14.6
Reference configuration		81.80	85.40	86.40	82.25	86.05	86.90	—	14.2

Reading the table. Three groups separate cleanly. (i) *Objective*: J_2 and J_3 cost 9.44 and 4.17 pp, about twice the largest effect of anything else. (ii) *Breadth*: the fan-out M and cluster count C jointly set how many single-cluster candidates get scored and are the only knobs that ever *improve* on the reference configuration; note that $C = 16$ at $M = 8$ is *worse* than $C = 8$ at $M = 8$, because raising C without raising M shrinks the fraction of clusters examined. (iii) *Near-inert*: the cluster prior is exactly neutral (with $C = M = 8$ the top-M screen admits every cluster), and the member prior, spatial link, allocator mode (except sequential fill) and clustering algorithm all move the macro by ≤ 1.4 pp.

U Additional limitations

The exact-576 component ablation (Appendix T) perturbs one component at a time; the intervention form (gain-of-information versus knock-out) is isolated only on the LLaVA-1.5 SCREEN (Appendix P). Untested are top-M screening against unrestricted subset search, the kNN neighbourhood size n, beam or combination search over clusters, and joint perturbations, so component interactions remain unmeasured, and the shallow search makes the candidate generator more consequential than a deep search would. All backbones are in the 4B–8B class; at larger scales the reference prefill is proportionally more expensive and is untested. Exact-576 cells rest on 2,000 examples, so 8 of 24 paired intervals against the per-cell best heuristic contain zero, all at $k \geq 128$ (Appendix C); only four of the ten integrated selectors are architecture-comparable under exact-576; and the three-seed check speaks to determinism, not to robustness under resampling. The held-out check of Appendix D covers two backbones and two budgets. Finally, no disaggregated evaluation is reported: because the criterion is agreement with the deployed model's own distribution, any bias in that distribution is what the selector optimises towards.

V Qualitative examples



Figure 7: Measured token-selection masks (1 of 2): object hallucination, counting and spatial relation. Each row is one SEED-Bench case (three per question type); columns show the original image and the tokens kept by CIG, MMTok and CDPruner at $k = 32$ and by FastV at $k = 64$ (30 tokens kept after its layer-2 prune). Masks are the selected indices recorded during evaluation.

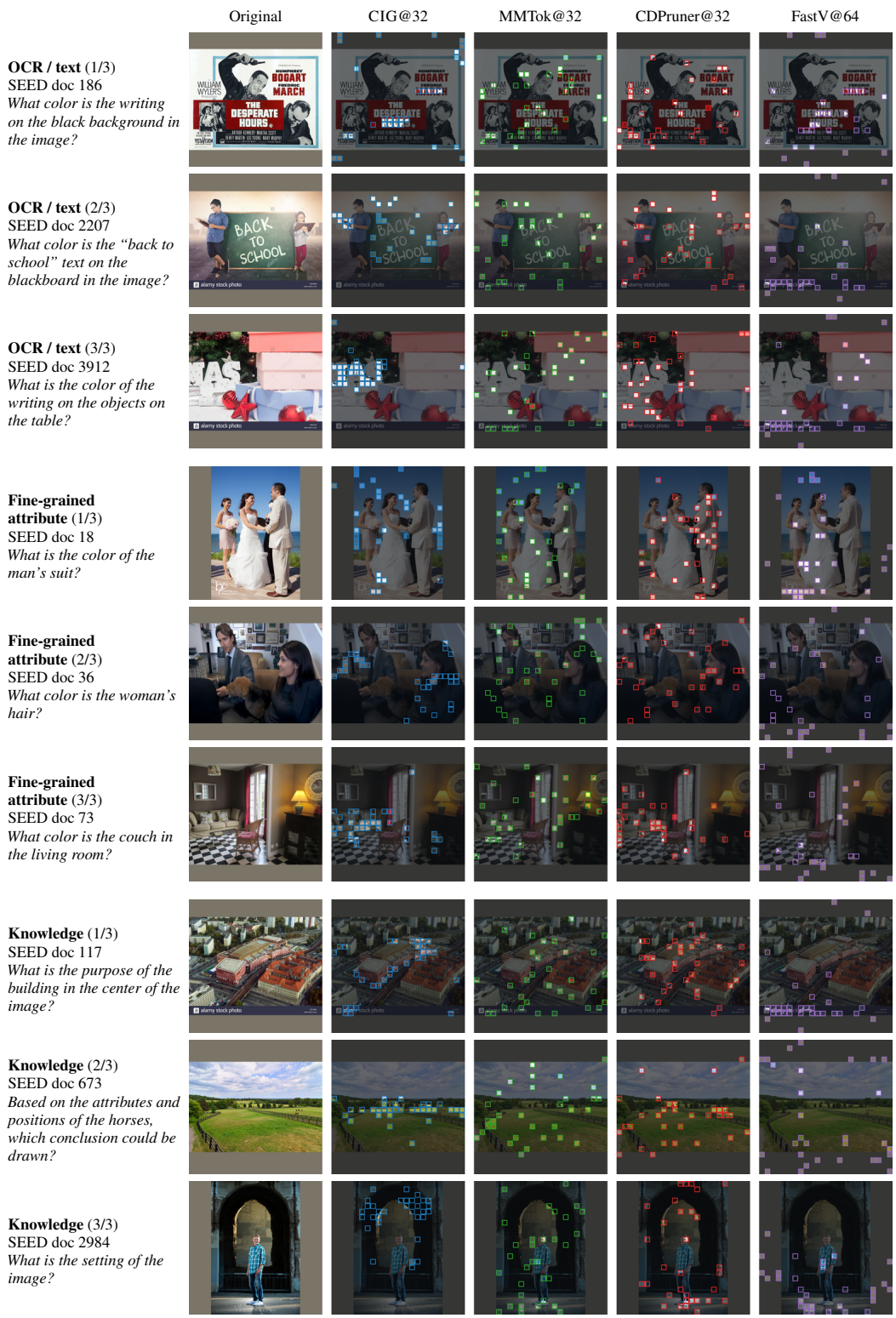


Figure 8: Measured token-selection masks (2 of 2): OCR/text, fine-grained attribute and knowledge; layout as in Figure 7. Where the OCR and knowledge categories had fewer than three net-gain cases on the screening split, the earliest matched cases from the full split are used.