SHAPRUNE: SHAPLEY-SUPERVISED VISUAL-TOKEN PRUNING FOR FROZEN VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We hypothesize that a frozen vision–language model's own removal effects identify the visual tokens it needs better than the attention it assigns them. To test this, we compute offline the Shapley values of the 64 3×3 visual-token blocks of LLaVA-1.5-7B on its own first answer token for 1,400 VQAv2 pairs, and amortize them into 256 weights, one per head of the first eight LLM layers, over each token's text-to-image attention. After layer 8 the scores select tokens by top-k (SHAPRUNE) or serve as the relevance of a determinantal point process (DPP; SHAPRUNE-D); inference evaluates no coalition. With depth, features and top-k fixed, the readout keeps 95.9/84.6% of POPE/GQA answers at 64 tokens, against 83.2/70.0% for FastV's score and 91.4/78.3% for the best label-free readout, and a registered fresh-image test confirms the gap to FastV's score (85.3 against 71.4%). As DPP relevance it keeps more answers than CLIP relevance and than none on 10,000 fresh images (registered; 88.01 against 86.58 and 85.45%; accuracy against CLIP relevance: no difference detected). Its 256 head-level weights keep more answers than a per-layer weighting trained on the same labels (registered), and cheaper removal-effect labels, down to single ablations, train heads from which no difference is detected. On 22 of 24 benchmark cells no difference from CLIP relevance at the same depth is detected, and at equal kept tokens a substantial part of SHAPRUNE-D's margin over earlier diversity-based pruners comes from pruning after layer 8. The model's own removal effects are thus a better pruning signal than its attention scores.

1 INTRODUCTION

A vision–language model such as LLaVA-1.5 (Liu et al., 2024) spends 576 of its input positions on one image, and most of them can be dropped. Pruners choose the tokens to keep by the attention they receive inside the LLM (Chen et al., 2024; Zhang et al., 2025c; Xing et al., 2025) or in the vision encoder (Yang et al., 2025), or by feature diversity: task-agnostic in DivPrune (Alvar et al., 2025), and in CDPruner (Zhang et al., 2025b) a conditional determinantal point process (DPP; Kulesza & Taskar, 2012) whose relevance is CLIP's similarity between each token and the question. Our hypothesis is that the frozen model's own removal effects, how much its answer moves when a group of visual tokens is taken away, identify the tokens it needs better than the attention it assigns them.

To test this, we measure the removal effects offline as Shapley values of 3×3 token blocks on the model's log-probability of the first token of its own un-pruned answer, for 1,400 VQAv2 pairs, and amortize them into 256 weights, one per head of the first eight LLM layers, over each token's text-to-image attention. At inference no coalition is evaluated: layers 0–7 run once, the readout scores the blocks, and the pruner keeps the top-k blocks (SHAPRUNE) or passes the scores as the relevance of CDPruner's greedy DPP (SHAPRUNE-D); layers 8–31 run on the kept tokens (Figure 1).

We evaluate on LLaVA-1.5-7B with POPE, GQA, MME and TextVQA against six published pruners, in four steps. *Removal effects beat attention scores* (§5.1): with depth, features and top-k rule fixed, the Shapley-supervised readout keeps 95.9/84.6% of POPE/GQA answers at 64 tokens, against 83.2/70.0% for FastV's score and 91.4/78.3% for the best label-free readout, and on 5,000 fresh images (registered) it keeps 85.3% against 71.4% for FastV's corrected score. *As a DPP relevance, it keeps more answers on fresh images* (§5.2): on 10,000 fresh images (registered), SHAPRUNE-D keeps more answers than CDPruner's CLIP relevance and than a DPP without relevance at the same

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.

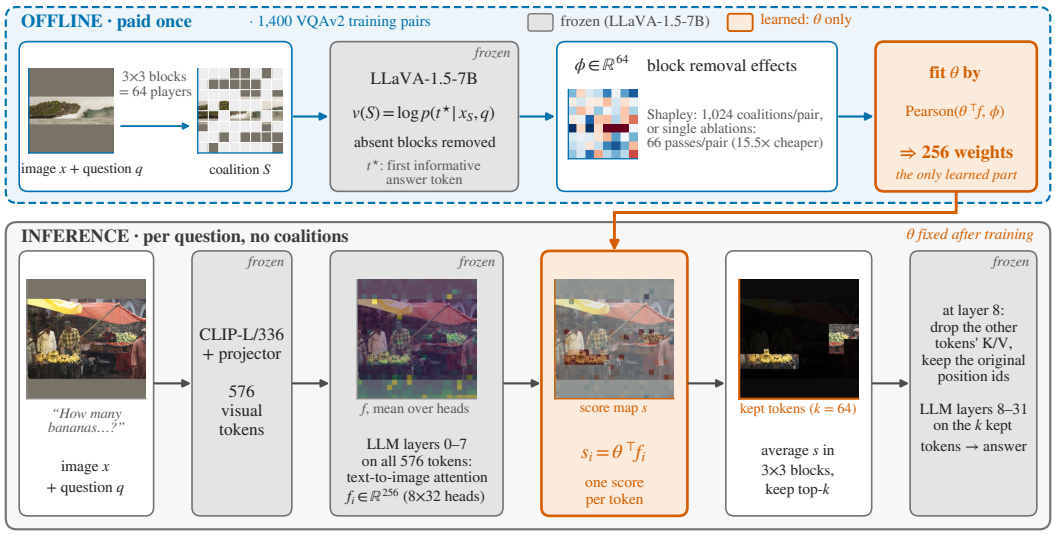

Figure 1: How the Shapley supervision is amortized and used. Grey: the frozen LLaVA-1.5-7B; only θ, 256 head weights, is learned. *Offline, once (top):* for each of 1,400 VQAv2 training pairs, the removal effects $\phi \in \mathbb{R}^{64}$ of the 64 3×3 token blocks on $v(S) = \log p(t^\star \mid x_S, q)$ are computed as Shapley values from 1,026 forward passes per pair (or as single ablations from 66); θ is fitted by maximizing the Pearson correlation of the block-averaged $\theta^\top f$ with ϕ. *Inference (bottom):* layers 0–7 run on all 576 visual tokens; each token's text-to-image attention in 8 layers × 32 heads, $f_i \in \mathbb{R}^{256}$, gives $s_i = \theta^\top f_i$; the figure shows top-k selection (SHAPRUNE), and SHAPRUNE-D replaces its top-k step by CDPruner's greedy DPP with the block scores as relevance (§3.4). Layers 8–31 run on the kept tokens.

depth (88.01 against 86.58 and 85.45%; accuracy against CLIP relevance: no difference detected). *SHAPRUNE-D beats earlier pruners at equal kept tokens* (§5.3): it keeps more answers than pruners acting earlier in 71 of 72 recall comparisons (post hoc), a substantial part of the margin over diversity-based pruners coming from its placement after layer 8. *The readout learns each pair's own removal effects, head by head, and the labels can be made cheap* (§5.4): the 256 head-level weights keep more answers than an 8-weight per-layer head trained on the same labels (registered), and re-solved coalitions, a Beta(16,1) semivalue and single ablations (15.5 times fewer forward passes) train heads from which no difference is detected in the deciding cells. We therefore consider the model's own removal effects, used to supervise token importance, a better pruning signal than its attention scores. Section 6 states where the method ties or loses and which choices were made post hoc.

2 Related work

Shapley attribution, amortization and attribution as supervision. KernelSHAP (Lundberg & Lee, 2017) estimates Shapley values by weighted regression; FastSHAP (Jethani et al., 2022) and ViT-Shapley (Covert et al., 2023) amortize them, and noisy labels suffice (Covert et al., 2024). RepoShapley (Huo et al., 2026) and Shapley Context Pruning (Chen et al., 2026) prune context by Shapley values, ContextCite (Cohen-Wang et al., 2024) fits a surrogate to context ablations of a model's own response, AT2 (Cohen-Wang et al., 2025) learns which attention heads to trust for text attribution, and vStream (Kang et al., 2026) fits one weight per (layer, head) of a VLM's text-to-image attention to predict the effect of masking a region. Our readout is AT2's recipe in vStream's setting, a far smaller amortizer than FastSHAP's, used as a pruning signal.

Visual-token pruning. Training-free methods rank tokens by attention inside the language model (Chen et al., 2024; Zhang et al., 2025c; Xing et al., 2025; Ye et al., 2025a) or in the vision encoder (Yang et al., 2025; Zhang et al., 2025a), or maximize diversity or remove duplicated tokens (Alvar et al., 2025; Zhang et al., 2025b; Wen et al., 2025b); VScan (Zhang et al., 2026) also prunes at an intermediate LLM layer and CAPA (Jha & Kim, 2026) scores attention times value vectors during

decoding; random subsets are strong baselines (Wen et al., 2025a; Wang et al., 2026). HAWK (Zhu et al., 2026b) weights first-layer heads by the benchmark drop when a head is masked. Learned selectors are trained on saliency (Tao et al., 2025), the task loss (Ye et al., 2025b; Huang et al., 2025; Takezoe et al., 2026; Zhu et al., 2026a; He et al., 2026), searched keep masks (Song et al., 2026), middle-layer attention (Sun et al., 2026b), gradient-weighted relevance (Lei et al., 2026) or, by reinforcement learning, the reference answer's log-probability (Shao et al., 2025); concurrent work weights heads per input without supervision (Ma et al., 2026; Sun et al., 2026a). None supervises its selection with per-block removal effects on the model's own answer, and none uses such a signal as the relevance of a diversity selector (HAWK-style weights on our features keep fewer answers than uniform weights; Appendix F). Appendix N compares the closest methods and covers ViT token reduction, visual-token withdrawal and the attention-as-explanation debate.

3 Method

LLaVA-1.5-7B encodes an image into $D = 576$ visual tokens that precede the question; a pruner keeps $k \in \{192, 128, 64\}$ of them. The model is frozen throughout.

3.1 A removal game on the model's own answer

For an image x and question q, let $a^\star$ be the frozen model's greedy answer and $t^\star$ its first answer token (for the training labels the first *informative* token, e.g. the digit rather than the word-boundary token before numerals). A coalition S of visual tokens has value

$$v(S) = \log p\big(t^\star \mid x_S,\, q\big), \tag{1}$$

with the tokens outside S removed from the sequence, not masked (one of several choices; Covert et al., 2021), and positions re-indexed as in pruning before the LLM. The Shapley value $\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(D-|S|-1)!}{D!}\,[v(S \cup \{i\}) - v(S)]$ averages a player's marginal contribution over coalitions. Selecting by ϕ needs $a^\star$: it is a reference, not a method (§5.1, Appendix C), and not an optimal selection when tokens interact; we use ϕ as a training signal and judge the scorer by the answers kept.

3.2 Shapley labels on blocks

With the 576 tokens as players, KernelSHAP estimates at 1,024 coalitions are not reproducible (split-half $+0.11$; Appendix E), so the players are the 64 non-overlapping 3×3 blocks of the 24×24 grid: 512 coalitions drawn from the Shapley kernel, each paired with its complement, give 1,024 coalition evaluations per pair (1,026 forward passes) and block values, solved in the efficiency-constrained closed form (Lundberg & Lee, 2017; Covert & Lee, 2021), with split-half reliability $+0.52$. We label 1,400 image–question pairs from VQAv2 train2014, whose COCO images are disjoint from POPE's and GQA's (120 held out for checkpoint selection) — about 1.4M forward passes, paid once. Single-ablation labels, $\delta_i = v(N) - v(N \setminus \{i\})$ from 66 forward passes per pair, are cheaper removal-effect labels that do not use the Shapley weighting (§5.4).

3.3 Amortizing the labels into a readout of early-layer attention

Let $A^{(l,h)}$ be the attention of head h in LLM layer l for the un-pruned sequence. For visual token i we average $A^{(l,h)}_{t,i}$ over the text positions t after the image, one number per (layer, head); with the first $K = 8$ of 32 layers and 32 heads each, token i has features $f_i \in \mathbb{R}^{256}$ and score

$$s_i = \theta^\top f_i + b, \qquad \theta \in \mathbb{R}^{256}. \tag{2}$$

As in vStream, θ is fitted from AT2's start $\theta = \mathbf{1}/256$ (the mean over all heads) by maximizing, within each training pair, the Pearson correlation between the block-averaged scores and the block Shapley values (validation Pearson $+0.28$ on the 120 held-out pairs); b is not identified and changes no selection (Appendix H). The labels exist only in training; the attention scores the pruner reads are the ones the model computes anyway (materializing them costs about 4.3 ms; §5.5).

3.4 Selection: top-k or the relevance of a conditional DPP

The scores are averaged within each 3×3 block into $\bar{s}_i$. SHAPrune keeps the k tokens of largest $\bar{s}_i$, ties inside the last, partly kept block broken by a permutation seeded per question (Appendix A). SHAPrune-D instead passes $\bar{s}$ to CDPruner's conditional DPP (Zhang et al., 2025b): with $\hat{f}_i$ the normalized projector output of token i and r the min–max-normalized relevance, the kernel is $L_{ij} = r_i\, \hat{f}_i^\top \hat{f}_j\, r_j$, and a greedy MAP of $\det L_S$ (Chen et al., 2018) adds one token per step, trading relevance against redundancy. CDPruner takes r from CLIP's token–question similarity; pure DPP sets r constant. The DPP is our implementation of CDPruner's loop, not bit-identical to its official code; CDPruner@8 runs the same code with CDPruner's own relevance and keeps the official token set on every question, so SHAPrune-D and CDPruner@8 differ only in the relevance.

3.5 Placement and cost

We prune once layers 0–7 have run on all tokens, as FastV does after layer 2 ('after layer K': after the first K layers, numbered from 0): layers 8–31 run only on the kept tokens with their original position ids. The LLM processes $8{\cdot}576{+}24k$ visual token-layers, as many as pruning to $k{+}(576{-}k)/4$ tokens before the LLM (192 for $k = 64$), at decoder FLOPs equal within 0.8% (Appendix G). The game removes tokens from the input, while in the deployed pruner layers 0–7 see all 576 tokens, so the game is a proxy: single-ablation labels recomputed under the deployed edit agree with the input-removal ones at a Pearson ratio of 0.75 (our registered criterion; Spearman ratio, computed afterwards, 0.50), and retraining a 350-pair head on them detects no recall change at 64 tokens (Appendix I).

4 Experimental setup

Benchmarks and baselines. POPE test (Li et al., 2023) (9,000 questions; accuracy and mean F1), GQA testdev-balanced (Hudson & Manning, 2019) (12,578), MME (2,374; perception + cognition, official script) and TextVQA validation (5,000; official evaluator), greedy decoding with the short-answer prompt (Appendix A). FastV (Chen et al., 2024) prunes after layer 2; VisionZip (Yang et al., 2025), CDPruner and DivPrune (Alvar et al., 2025) before the LLM; SparseVLM (Zhang et al., 2025c) and PyramidDrop (Xing et al., 2025) progressively inside it (PyramidDrop at the schedule its authors recommend for about 64 tokens per layer). All run from official code (Appendices J and K). We add pure DPP and a matched random subset, seeded per question, and also run FastV, pure DPP, CDPruner and matched random after layer 8 to separate the placement from the signal.

Answer recall and tests. *Answer recall* is the fraction of questions on which the pruned model's answer equals the un-pruned model's (exact match after lower-casing), a fidelity measure; we report accuracy beside it but do not rank selectors by accuracy, because pruning can move a model's answers toward a benchmark's label balance: CDPruner's POPE F1 at 64 tokens, 87.5, exceeds the un-pruned model's 85.9 (Appendix B). Two methods are compared on the same questions by an exact McNemar test; POPE's 9,000 rows hold 5,170 distinct image–question pairs, and its tests count each once. A difference is claimed only at $p < 0.01$ with a 99% image-level bootstrap interval excluding zero; everything else is a *tie*, meaning no detected difference, not equivalence (no margin was fixed). Registration status is stated where each comparison appears (history: Appendix A).

5 Results

5.1 Shapley-supervised importance against attention scores

We first test the hypothesis directly; outside TextVQA it holds. Figure 2 fixes the placement after layer 8, the 256 attention features and the top-k rule, and, FastV and pure DPP aside, changes only the map from features to one score, so these gaps are the signal's, not the placement's. With uniform θ (all heads' mean attention) the pruner keeps 67.0% of POPE and 58.8% of GQA answers at 64 tokens, below matched random, and its POPE yes-rate collapses to 10.2% against the un-pruned 42.2%, consistent with the bottom-of-image bias of early-layer attention (Endo et al., 2025; Zhao et al., 2025). Layers 6–7 alone give 83.8 and 69.5%, FastV's own score at layer 8, 83.2 and 70.0%, and the best label-free readout we built (position-debiased, layers 6–7; registered) 91.4 and 78.3%.

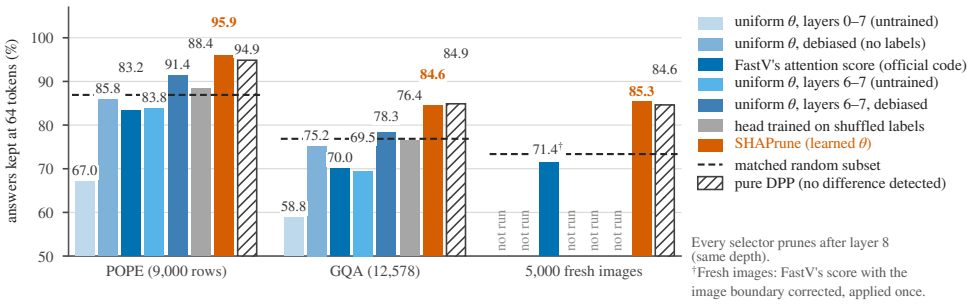

Figure 2: Shapley supervision against attention scores at the same depth: *answer recall* at 64 tokens, every selector after layer 8, and top-k except pure DPP. The uniform-θ readouts (two debiased without labels), the shuffled-label head and SHAPRUNE read the same 256 attention features and differ only in the map to one score; FastV uses its own score; pure DPP reads no attention; dashed: matched random. Tests (POPE on 5,170 distinct pairs): §5.1, Table 26; accuracies, 192 and 128 tokens (not for the debiased readouts): Table 12.

The Shapley-supervised weights keep 95.9 and 84.6% ($p < 10^{-15}$ against each). Against every attention-score readout we ran, SHAPRUNE keeps more answers at 64 tokens in every POPE, GQA and MME cell and in every TextVQA cell but one, by 4.5 to 34.5 POPE points on the 9,000 rows (3.0 to 21.4 on the distinct pairs; Table 7, Appendix F): the information is in the attention, and attention scores underuse it. FastV's configuration does not explain its gap: with its image boundary corrected and its score applied once, it still loses every POPE, GQA and MME cell at 64 tokens (Table 21), and SparseVLM and PyramidDrop lose all 12 and 4 cells at their nominal budgets, although PyramidDrop's recommended schedule does not reproduce its published numbers (Appendix K). TextVQA, with multi-token answers, is the exception (§6).

Confirmation on fresh images. The head and the placement were chosen on development data and full-set results, so we froze them and registered a test on 5,000 VQAv2 questions, one per image, on COCO images used nowhere else (Appendix M). All three hypotheses on *answer recall* at 64 tokens came out as registered (Table 26): SHAPRUNE keeps more answers than FastV's corrected score applied once (H1: 85.3 against 71.4%, $[+12.2, +15.6]$) and than matched random (H3: 73.4%, $[+10.3, +13.6]$), both at layer 8, and no difference is detected against pure DPP@8 (H2: 85.3 against 84.6%, $[-0.6, +2.1]$). Outside the hypotheses, CDPruner@8 is more accurate at 64 tokens (69.5 against 68.4%, $p = 0.008$).

Headroom: the Shapley reference. Selecting the 64 tokens of highest answer-aware token-level Shapley value keeps LLaVA's answer on all 99 POPE questions and 95% of 200 GQA questions, against 92% and 84% for CDPruner; SHAPRUNE keeps 94.9 and 87.5% (descriptive), with no difference detected from CDPruner on these questions, and the reference stays ahead on GQA (22 questions against 7, $p = 0.008$; Appendix C). The reference needs the answer and is not a bound, and answer-aware single ablation ties it wherever ablation was computed, so the headroom is for answer-aware selection, not a property of Shapley values (§6).

5.2 THE SHAPLEY-SUPERVISED READOUT AS THE RELEVANCE OF A DPP

As a DPP relevance, the signal keeps more answers than CLIP relevance and than none in a registered 10,000-image test (accuracy against CLIP relevance: no difference detected). We devised SHAPRUNE-D after CDPruner@8 kept more GQA answers than top-k SHAPRUNE (§6) and registered its comparison with CDPruner@8 before its runs. Table 1 holds the depth fixed and varies the relevance. Against attention scores (FastV@8) SHAPRUNE-D wins 9 of 12 recall and 7 of 12 accuracy cells (losing TextVQA recall at 192 tokens), and on the first fresh set keeps 87.16 against 71.40% for FastV's corrected score applied once; against pure DPP@8 it wins 7 of 24 cells (Table 2), among them POPE and GQA recall at 64 tokens (GQA 86.6 against 84.9%), and keeps more answers on the first fresh set (87.16 against 84.62%, $p = 1.7 \times 10^{-7}$; the fresh-set comparison post hoc, the cells registered as descriptive); it loses no cell to pure DPP@8, CDPruner@8 or SHAPRUNE. Rele-

Table 1: SHAPRUNE-D after layer 8 against the other selections at its depth: *answer recall* verdicts (W/T/L, win/tie/loss for SHAPRUNE-D) over the 12 benchmark–budget cells of POPE, GQA, MME and TextVQA and the 12 accuracy cells, and *answer recall* at 64 tokens on out-of-sample sets (SHAPRUNE-D / comparator, %, verdict). Post hoc unless marked registered (R), except the cells against pure DPP@8 and SHAPRUNE, registered as descriptive; for CDPruner@8 the four recall cells at 64 tokens decide, the rest were registered for reporting only. First fresh set: 5,000 VQAv2 questions on unused COCO images; VizWiz: 4,319 questions; G5: 10,000 further fresh images. [c]Accuracy: no difference detected (70.47 / 70.46%). [a]FastV's corrected score applied once. [b]POPE and GQA only. A dash: not run. Per-cell values: Table 2.

Comparator at layer 8	Relevance / diversity	Recall (12)	Acc. (12)	First fresh set	VizWiz	G5 (R)
CDPruner@8	CLIP / yes	2/10/0 (R: 4 at 64)	0/12/0	87.16 / 86.12 T (R)	79.88 / 79.67 T (R)	88.01 / 86.58 W[c]
Pure DPP@8	none / yes	6/6/0	1/11/0	87.16 / 84.62 W	79.88 / 78.98 T (R)	88.01 / 85.45 W
SHAPRUNE (top-k)	learned / no	8/16/0 over the 24 cells		87.16 / 85.34 W	79.88 / 77.12 W (R)	–
FastV@8	attention score / no	9/2/1	7/5/0	87.16 / 71.40[a] W	–	–
Matched random@8	none / no	6/0/0[b]	6/0/0[b]	87.16 / 73.36 W	–	–

Table 2: SHAPRUNE-D (the head's block scores as the DPP relevance) after layer 8, a variant proposed after the CDPruner@8 result of Table 10. Columns 4–8: against CDPruner@8, the registered comparison; POPE rates and tests on the 5,170 distinct pairs, not the 9,000 rows of Table 3; w–ℓ: questions only SHAPRUNE-D / only CDPruner@8 gets right; 99% image-level interval of SHAPRUNE-D minus CDPruner@8, in points. †: SHAPRUNE-D keeps more answers; every other cell is a tie (no difference detected). Last three columns, descriptive and not claims: verdict of SHAPRUNE-D against pure DPP at layer 8 (DPP@8), SHAPRUNE at layer 8 (S@8) and SHAPRUNE-D before the LLM (D-pre), by the same rule; W win, T tie. Fresh images, recall at 64 tokens against CDPruner@8: 87.16 against 86.12%, a tie. Post hoc equal-kept-token tally: Appendix O.

Bench.	k	Metric	SHAPRUNE-D	CDP@8	w–ℓ	p	99% interval	DPP@8	S@8	D-pre
POPE	192	recall	98.30	98.20	28–23	0.58	$[-0.27, +0.47]$	W	T	W
POPE	192	acc.	89.75	89.61	29–22	0.4	$[-0.23, +0.51]$	T	T	T
POPE	128	recall	97.56	97.39	44–35	0.37	$[-0.27, +0.63]$	T	T	W
POPE	128	acc.	89.90	89.50	50–29	0.024	$[-0.04, +0.90]$	T	T	T
POPE	64	recall	96.71	96.46	76–63	0.31	$[-0.37, +0.83]$	W	T	W
POPE	64	acc.	89.71	89.34	79–60	0.13	$[-0.21, +0.95]$	T	T	T
GQA	192	recall	93.22	93.06	342–322	0.46	$[-0.40, +0.75]$	W	W	W
GQA	192	acc.	61.40	61.40	198–198	1	$[-0.45, +0.46]$	T	T	W
GQA	128	recall	90.97	90.83	478–460	0.58	$[-0.52, +0.81]$	W	W	W
GQA	128	acc.	61.33	61.31	308–306	0.97	$[-0.52, +0.55]$	T	T	W
GQA	64	recall	86.60	86.93	645–686	0.27	$[-1.09, +0.46]$	W	W	W
GQA	64	acc.	60.52	60.68	444–464	0.53	$[-0.79, +0.47]$	T	T	W
MME	192	recall	96.38	96.12	25–19	0.45	$[-0.46, +1.01]$	T	T	W
MME	192	acc.	79.15	79.40	19–25	0.45	$[-0.97, +0.46]$	T	T	W
MME	128	recall	94.52	94.90	38–47	0.39	$[-1.43, +0.63]$	T	T	W
MME	128	acc.	79.19	78.98	45–40	0.66	$[-0.76, +1.22]$	T	T	W
MME	64	recall	93.30	92.38	68–46	0.049	$[-0.21, +2.15]$	T	W	W
MME	64	acc.	78.90	78.05	67–47	0.075	$[-0.34, +2.06]$	T	T	W
TextVQA	192	recall	88.58	87.32	224–161	1.5×10^{-3}	$[+0.22, +2.30]^{\dagger}$	W	W	W
TextVQA	192	acc.	58.54	57.76	71–43	0.011	$[+0.17, +1.38]$	W	T	T
TextVQA	128	recall	85.48	84.64	246–204	0.053	$[-0.28, +1.96]$	T	W	W
TextVQA	128	acc.	57.98	57.56	79–66	0.32	$[-0.26, +1.08]$	T	T	T
TextVQA	64	recall	80.82	79.38	323–251	3.0×10^{-3}	$[+0.18, +2.71]^{\dagger}$	T	W	W
TextVQA	64	acc.	57.01	56.35	111–80	0.03	$[-0.10, +1.49]$	T	W	T
W/T/L over the 24 cells (vs CDPruner@8: 2/22/0)								7/17/0	8/16/0	18/6/0

vance and diversity keep different answers: top-k SHAPRUNE and pure DPP@8 tie in rate but not in selection, since at 64 tokens only one of them keeps the answer to 4.8% of distinct POPE pairs and 15.5% of GQA questions, about eight times the rate (0.6 and 2.0%) at which two seeds of SHAPRUNE disagree (post hoc; Appendix A). At layer 8, CDPruner's relevance beats pure DPP in 5 of 24 cells, losing none.

Table 3: LLaVA-1.5-7B with 64 of 576 visual tokens. Every cell leads with *answer recall*; POPE F1 is the mean over its categories, MME's score is perception + cognition. Time: median time to first token over 300 requests on one RTX 3090 at batch size 1, reference selection code; –: not timed end to end. Tok./layer: mean visual tokens per decoder layer (Table 18); *token-dropping variant of FastV (the official code we ran masks); ‡second run (Table 4). Bold: best pruned recall, accuracy or F1. At 64 tokens SHAPRUNE-D keeps more answers than every row of the upper block it was tested against and than FastV at layer 8 (post hoc; SparseVLM and PyramidDrop were not tested against it); against CDPruner@8 it wins on TextVQA and ties elsewhere; against pure DPP@8 it wins on POPE and GQA. POPE rates are over its 9,000 rows, its tests over the 5,170 distinct pairs (rates: Table 2). MME's score is not tested per question (Appendix J). 192 and 128 tokens: Appendices F, J and O.

Method	POPE recall / acc / F1	GQA recall / acc	MME recall / score	TextVQA recall / acc	Tok./ layer	Time (ms)
Un-pruned (576 tokens)	100 / 87.0 / 85.9	100 / 61.9	100 / 1862.4	100 / 58.23	576	197.7
Matched random	87.2 / 79.5 / 75.4	74.1 / 55.3	82.4 / 1535.8	62.0 / 48.27	64	–
FastV (layer 2)	78.7 / 70.8 / 59.2	67.8 / 51.5	85.5 / 1557.5	76.4 / 55.01	96*	–
SparseVLM	88.0 / 81.0 / 77.6	72.1 / 53.7	84.5 / 1600.6	69.2 / 53.46	82	–
PyramidDrop (64)	71.0 / 63.4 / 42.7	57.6 / 45.0	73.5 / 1255.5	61.2 / 47.82	63	–
VisionZip	88.5 / 80.6 / 77.0	75.6 / 55.1	87.5 / 1687.6	73.6 / 55.55	64	–
CDPruner	94.7 / **87.8** / **87.5**	82.2 / 58.6	87.1 / 1709.8	73.1 / 55.14	64	98.3
Pure DPP (0 param.)	94.4 / 87.3 / 86.5	80.8 / 58.3	87.8 / 1683.9	74.3 / 55.63	64	90.0
DivPrune	93.9 / 86.5 / 85.5	80.6 / 57.7	86.6 / 1615.4	73.3 / 54.55	64	–
After layer 8						
FastV (official)	83.2 / 75.0 / 67.5	70.0 / 52.6	85.4 / 1527.7	78.2 / 55.07	192*	–
Pure DPP@8 (diversity)	94.9 / 86.1 / 84.9	84.9 / 60.0	92.2 / 1733.6	79.6 / 56.37	192	120.0
CDPruner@8 (CLIP + div.)	96.1 / 87.1 / 86.4	**86.9** / **60.7**	92.4 / 1778.1	79.4 / 56.35	192	129.0
SHAPRUNE (Shapley, top-k)	95.9 / 86.6 / 85.6	84.6 / 60.1	91.9 / 1850.7	79.0 / 56.08	192	117.6
SHAPRUNE-D (Shapley + div.)	**96.4** / 87.1 / 86.3	86.6 / 60.5	**93.3** / 1839.4	**80.8** / **57.01**	192	128.4‡

A registered test on 10,000 new images (G5). Before building the data we registered a test on 10,000 further fresh VQAv2 images, disjoint from the first fresh set and from all training and pair-list images: *answer recall* at 64 tokens after layer 8 and the rule of §4 decide E1, SHAPRUNE-D against CDPruner@8, and E2, SHAPRUNE-D against pure DPP@8. SHAPRUNE-D keeps more answers than CDPruner@8 (88.01 against 86.58%, 503–360, $p = 1.3 \times 10^{-6}$, $[+0.67, +2.16]$), with no accuracy difference detected (70.47 against 70.46%, $p = 1$), and than pure DPP@8 (88.01 against 85.45%, 662–406, $p = 4.5 \times 10^{-15}$, $[+1.73, +3.39]$). Pooled post hoc over both fresh sets (15,000 images), SHAPRUNE-D keeps 87.73 against 86.43% for CDPruner@8 ($[+0.66, +1.90]$; accuracy: no difference detected) and 85.17% for pure DPP@8. Only these two cells are confirmatory; the earlier, smaller tests' ties with CDPruner@8 stand (§6).

5.3 Equal kept tokens against published pruners

At equal kept tokens SHAPRUNE-D keeps more answers than pruners acting earlier (post hoc): at 64 tokens (Table 3) it keeps more un-pruned answers than every pruner acting before the LLM or at layer 2 that we tested against it, on every benchmark, and more than FastV at layer 8. Over the 114 recall cells of our comparison axis (six pruners at their own positions and four at layer 8, three budgets), SHAPRUNE-D wins 94, ties 19 and loses 1 (post hoc; Appendix O): 71, 1 and 0 of the 72 against pruners acting earlier (the tie: FastV at layer 2 on TextVQA at 192 tokens) and 23, 18 and 1 of the 42 at its own depth. On accuracy, over the same 114 cells, it wins 69, ties 45 and loses none (post hoc); on POPE, where CDPruner before the LLM is numerically more accurate over the 9,000 rows (87.8 against 87.1% at 64 tokens), the test on the distinct pairs detects no difference; against SHAPRUNE, a post hoc test on all 9,000 POPE rows finds CDPruner more accurate at every budget (by 0.89 to 1.29 points; Appendix F). The registered tally of top-k SHAPRUNE is 83 won, 26 tied and 5 lost, all losses at its own depth. These comparisons hold kept tokens equal, not placement: SHAPRUNE-D prunes after layer 8, and a substantial part of the margin over diversity-based pruners comes from that placement (§6, Appendix F).

5.4 What the supervision needs: paired, reproducible removal effects

Finally, the readout learns each pair's own removal effects, head by head, from labels that can be made cheap. *The supervision carries the signal.* The identical head trained on labels permuted across pairs

Table 4: Measured end-to-end latency: median time to first token (ms) over 300 requests, one RTX 3090, batch size 1; selection-stage medians (ms); peak memory above the weights over a request (MiB; maximum over the two repeats, then median over requests). Reference: the selection code of the evaluated runs; optimized: a GPU version selecting the same tokens. SHAPRUNE: eager path (the fused variant failed its token-set gate; Appendix G). SHAPRUNE-D rows: a second run (same protocol), in which the pruned pipelines shared with the first run were 2.8 to 6.6 ms slower (un-pruned 15.2); the text compares SHAPRUNE-D with others only by differences paired within that run.

	Time to first token			Selection stage			Memory
Pipeline	64	128	192	64	128	192	(MiB)
Un-pruned (576)		197.7			–		381
SHAPRUNE after layer 8	117.6	129.4	140.1	0.8	0.8	0.8	234
Pure DPP after layer 8, reference	120.0	137.4	154.9	7.4	14.0	20.7	150–154
Pure DPP after layer 8, optimized	114.6	127.9	139.9	1.8	3.3	4.9	150–154
CDPruner after layer 8, reference	129.0	153.8	178.6	14.8	28.9	42.9	150–153
CDPruner after layer 8, optimized	116.7	130.0	142.1	1.9	3.5	5.1	150–153
SHAPRUNE-D after layer 8, reference	128.4	146.1	163.9	8.1	14.8	21.4	235
SHAPRUNE-D after layer 8, optimized DPP	122.8	136.3	148.7	2.6	4.1	5.8	235
CDPruner before the LLM, reference	98.3	126.7	154.2	14.8	29.0	43.1	106–155
CDPruner before the LLM, optimized	87.1	104.9	120.5	1.9	3.5	5.2	106–155
Pure DPP before the LLM, reference	90.0	111.5	131.4	7.3	13.9	20.5	109–155
Pure DPP before the LLM, optimized	85.0	102.6	118.6	1.8	3.3	4.9	109–155

keeps 88.4% of POPE and 76.4% of GQA answers and only ties matched random at the same depth, and a head trained on the mean label map (registered) keeps 86.0 and 75.4% (Appendix I): what the head learns is each image–question pair's own removal effects. *Blocks make the labels reproducible* (split-half $+0.52$ against $+0.11$ for tokens; §3.2). *The labels can be made cheap* (shown with top-k selection). Other removal-effect labels train heads with no detected difference from SHAPRUNE: difference-of-means and least-squares re-solutions of the same coalitions tie SHAPRUNE in every recall cell and eleven of twelve accuracy cells (Table 14), a Beta(16,1) semivalue (Kwon & Zou, 2022) changes no decision cell detectably (registered; Appendix I), and single-ablation labels (15.5 times fewer forward passes; 92K instead of 1.4M), registered as their own test with POPE recall at 64 tokens deciding, show no detected difference (SHAPRUNE 95.9 against 96.3% on the 9,000 rows; on the distinct pairs $p = 0.12$, $[-0.81, +0.17]$), and in all twelve cells are never significantly better, while SHAPRUNE is better in one, GQA recall at 192 tokens (92.3 against 91.7%). A second training seed differs in none of 12 cells (Appendix F); heads reading only the first 2, 4 or 6 layers tie matched random, and head-level weights add to a per-layer weighting learned from the same labels (registered; recall at 64 tokens, POPE 96.3 against 95.4% on distinct pairs, GQA 84.6 against 81.8%; Appendix I).

5.5 COST

Measured end to end (Table 4; differences below are paired per-request means, not differences of medians, and their 99% intervals exclude zero), the head's selection stage takes 0.8 ms at every budget plus about 4.3 ms to materialize the attention it reads; no CLIP text pass is needed. Paired within the second timing run, SHAPRUNE-D's reference form is 4.9 to 20.2 ms faster to the first token than CDPruner@8's reference ($[-21.0, -4.0]$) but 3.9 to 4.3 ms slower than pure DPP@8's ($[+3.1, +5.0]$), and needs SHAPRUNE's memory (235 against 150–154 MiB). Against SHAPRUNE, the reference DPP at layer 8 is 1.9 to 14.0 ms slower but the optimized DPP 0.6 to 3.2 ms faster (before the LLM: §6).

Equal kept tokens are not equal compute. Matched on visual token-layers (decoder FLOPs within 0.8%), a pruner acting before the LLM keeps 192 tokens where SHAPRUNE keeps 64 after layer 8 (Table 19). There SHAPRUNE beats matched random on all four benchmarks and FastV at layer 8 (Table 7) on POPE, GQA and MME, ties VisionZip, CDPruner, pure DPP and SparseVLM (5% more FLOPs) on POPE and MME, and loses on GQA to the last three and on TextVQA to all four; pure DPP at layer 8 fares almost exactly as SHAPRUNE does, so these losses come with pruning 64 tokens after layer 8. Post hoc, SHAPRUNE-D at 64 tokens ties CDPruner at 192 before the LLM in 7 of 8 recall and accuracy cells, losing GQA recall ($[-2.48, -0.71]$), ties pure DPP in all but a POPE recall win, and loses TextVQA recall to SparseVLM and no cell to the other pruners (Appendix G).

6 LIMITATIONS

Labels and same-depth baselines. Heads trained on difference-of-means, least-squares, Beta(16,1) or single-ablation labels show no detected difference from SHAPRUNE in the deciding cells (single ablation is never better, and worse in one of 12 cells, GQA recall at 192 tokens; §5.4). These controls used top-k selection, and SHAPRUNE-D's relevance was trained only on Shapley labels, so we do not claim that the Shapley weighting is necessary or better than other removal-effect labels; on the reference questions, answer-aware single ablation ties the Shapley reference (§5.1). At the same depth, top-k SHAPRUNE shows no detected difference from pure DPP@8 in any of 24 cells, CDPruner@8 keeps more GQA answers than it at every budget (86.9 against 84.6% at 64 tokens), and pure DPP@8 keeps more VizWiz answers (79.0 against 77.1%). SHAPRUNE-D ties CDPruner@8 in 22 of 24 benchmark cells, on the first fresh set (87.16 against 86.12%) and on VizWiz, and ties pure DPP@8 on VizWiz; its registered TextVQA win at 64 tokens ($p = 3.0 \times 10^{-3}$) would not survive Holm correction over its four decision cells and was not confirmed on fresh images (Tables 1, 2).

Placement, compute and latency. A substantial part of the margin over diversity-based pruners at equal kept tokens comes from the placement after layer 8, at up to three times their visual token-layers: pure DPP gains from layer 8 in 11 of 12 recall cells and CDPruner in 19 of 24. Post hoc, for SHAPRUNE, the move accounts for 83.3 to 111.9% of the recall margin over pure DPP in 11 of 12 cells and 94.1 to 199.0% of that over CDPruner in all 12; no positive signal part is detected in any cell, and over CDPruner it is negative on GQA (Appendix F). Before the LLM, the top-k head only ties CDPruner on POPE recall and trails it on GQA (Appendix J). Matched on visual token-layers, a pruner acting before the LLM keeps 192 tokens where ours keep 64; there SHAPRUNE loses on GQA and TextVQA to most such pruners (Table 19), and SHAPRUNE-D (post hoc) loses GQA recall to CDPruner (86.6 against 88.2%) and TextVQA recall to SparseVLM (Appendix G). Optimized, SHAPRUNE-D is 1.8 to 2.4 ms slower than CDPruner@8 and 4.7 to 5.1 ms slower than pure DPP@8 (paired within its second run). Pruning before the LLM is 4.4 to 33.9 ms faster than SHAPRUNE (except CDPruner's reference loop at 192 tokens) and uses less memory (106 to 155 against 234–235 MiB), since SHAPRUNE's layers 0–7 hold all 576 keys and values; time differences are paired per-request means, not differences of medians.

Backbone, design history and TextVQA. On Qwen2.5-VL-7B (noisier labels, a forced 24×24 grid), the head only ties matched random on POPE recall before the LLM, pure DPP keeps more POPE answers than it after layer 8 at 192 and 128 tokens, and SHAPRUNE-D was not run (Appendix L). The placement after layer 8 was designed after full-set GQA results and SHAPRUNE-D after the CDPruner@8 result; the equal-kept-token tally of SHAPRUNE-D is post hoc, and the head was chosen on POPE development questions. SHAPRUNE-D has one head seed, SHAPRUNE one head seed on MME and TextVQA, and the shuffled-label control one permutation; both fresh sets come from VQAv2, like our labels and part of LLaVA-1.5's instruction data. On TextVQA, FastV at layer 8 keeps more answers than SHAPRUNE at 192 and 128 tokens (89.8 against 87.7% and 85.8 against 84.2%) and than SHAPRUNE-D at 192 (89.8 against 88.6%); the game scores one token, not the answer string, and a whole-answer objective changes no TextVQA cell detectably. We ran every baseline ourselves. The labels cost about 1.4M forward passes, 58.76 GPU-hours (92K and 3.73 GPU-hours as single ablations; Appendix H), and must be redone for another backbone.

7 CONCLUSION

We hypothesized that a frozen model's own removal effects identify the visual tokens it needs better than its attention, and amortized block Shapley values on its own answer into 256 head weights over early-layer attention. At the same depth the readout keeps far more answers than FastV's score and every label-free readout we built, and as a DPP's relevance more than CLIP relevance and than none on 10,000 fresh images (registered; accuracy against CLIP relevance: no difference detected); it learns each pair's own removal effects head by head, and 15.5-times-cheaper single-ablation labels train it with no difference detected. At equal kept tokens a substantial part of the margin comes from the placement; we recommend 128 to 192 kept tokens, layer 8 when kept tokens bind, before the LLM when latency or memory does. The model's own removal effects are therefore a better pruning signal than its attention scores.

AI use statement

In this work, we used generative AI tools (a coding agent) to implement the methods, run the baselines, design and give feedback on experiments, propose and refine hypotheses when an experiment failed, interpret results, write and edit code, create figures, search and summarize the literature, draft and edit parts of this manuscript, translate our drafts and notes into English, clean and reformat datasets (converting benchmark and question files into each code base's input format, checked against the original files), and propose the title and keywords; we did not use them to generate synthetic datasets, formulate mathematical claims or write proofs (the paper contains no proofs), and neither theoretical models or conceptual frameworks nor qualitative or thematic data analysis is part of this work. We reviewed all AI-assisted work. Decision criteria were recorded in a lab journal before the corresponding results were seen, except where the paper states otherwise; the numbers in the paper were recomputed from per-question output files by scripts and checked against a ledger of results; code and manuscript changes were checked in review passes by separate agent instances that did not author them; and baselines were run from their official code. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics statement

All experiments use public benchmarks (VQAv2, COCO, POPE, GQA, MME, TextVQA) under their original terms; no human subjects were involved and no new data were collected. *Answer recall* rewards preserving the un-pruned model's answers, including its errors and biases: a pruner with higher recall is more faithful to the model, not more accurate or fairer, so accuracy is reported alongside it.

Reproducibility statement

The official code of VisionZip, FastV, CDPruner, DivPrune (commit 799e2d9), SparseVLM and PyramidDrop is used unmodified except for FastV's rotary-table length on 5 TextVQA prompts (Appendix J); each reproduction and its deviation from the published number, the largest being PyramidDrop's at an average of 64 tokens, is listed in Appendices J and K. Our code, the block labels, the trained heads and every method's per-question outputs can be shared anonymously with reviewers on request during the discussion period; seeds, splits and the paired-test procedure are in Appendix A.

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.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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.

Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. The struggles of feature-based explanations: Shapley values vs. minimal sufficient subsets. *arXiv preprint arXiv:2009.11023*, 2020.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy MAP inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

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.

Yanqiao Chen, Dongsheng Hou, Yuhan Rui, Zhen Cao, and Yepang Liu. Shapley context pruning: A cooperative game perspective for context reranking and pruning. *arXiv preprint arXiv:2607.16209*, 2026.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. ContextCite: Attributing model generation to context. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Benjamin Cohen-Wang, Yung-Sung Chuang, and Aleksander Madry. Learning to attribute with attention. *arXiv preprint arXiv:2504.13752*, 2025.

Ian Covert and Su-In Lee. Improving KernelSHAP: Practical Shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Ian Covert, Chanwoo Kim, and Su-In Lee. Learning to estimate Shapley values with vision transformers. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2206.05282.

Ian Covert, Chanwoo Kim, Su-In Lee, James Zou, and Tatsunori Hashimoto. Stochastic amortization: A unified approach to accelerate feature and data attribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2401.15866.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. arXiv:2407.09141.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. arXiv:2412.13180.

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

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

Peter L. Hammer and Ron Holzman. Approximations of pseudo-Boolean functions; applications to game theory. *Methods and Models of Operations Research (ZOR)*, 36(1):3–21, 1992.

Landi He, Xiaoyu Yang, and Lijian Xu. The model knows which tokens matter: Automatic token selection via noise gating. *arXiv preprint arXiv:2603.07135*, 2026.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, Yao Hu, and Shaohui Lin. Dynamic-LLaVA: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2412.00876.

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.

Yu Huo, Kun Zeng, Siyu Zhang, Yuquan Lu, Cheng Yang, Yifu Guo, and Xiaoying Tang. RepoShapley: Shapley-enhanced context filtering for repository-level code completion. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. arXiv:2601.03378.

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

Neil Jethani, Mukund Sudarshan, Ian Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time Shapley value estimation. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2107.07436.

Samyak Jha and Junho Kim. CAPA: Contribution-aware pruning and FFN approximation for efficient large vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 27543–27558, 2026. arXiv:2602.00247.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2503.03321.

Seil Kang, Woojung Han, Junhyeok Kim, Jinyeong Kim, Youngeun Kim, and Seong Jae Hwang. Real-time visual attribution streaming in thinking model. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2604.16587.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

Yongchan Kwon and James Zou. Beta Shapley: a unified and noise-reduced data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

Lei Lei, Jie Gu, Xiaokang Ma, Chu Tang, Jingmin Chen, and Tong Xu. Task-related token compression in multimodal large language models from an explainability perspective. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2506.01097.

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. arXiv:2305.10355.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations (ICLR)*, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025. arXiv:2405.05803.

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)*, 2024. arXiv:2310.03744.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Chaofang Ma, Lin Jiang, Carol Jingyi Li, Xingyu Liu, Zeyu Li, Jiang Xu, and Wei Zhang. Not all attention heads contribute to critical visual token selection: Head-aware pruning matters more. *arXiv preprint arXiv:2608.25332*, 2026.

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)*, pp. 8748–8763, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

Zhenwei Shao, Mingyang Wang, Weijun Zhang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, and Jun Yu. Growing a multi-head twig via distillation and reinforcement learning to accelerate large vision-language models. *arXiv preprint arXiv:2503.14075*, 2025. Extended version of the ICCV 2025 paper.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games, Volume II*, number 28 in Annals of Mathematics Studies, pp. 307–317. Princeton University Press, 1953.

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)*, pp. 8317–8326, 2019.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. EvoComp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2604.17087.

Yuanhao Sun, Huawei Ji, Yuan Jin, Cheng Deng, Luoyi Fu, and Xinbing Wang. HAP: Head-adaptive visual token pruning via cross-modal alignment. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026a. arXiv:2608.23921.

Yuyao Sun, Tao Deng, Shuang Li, Deqing Wang, Hao Geng, and Minjun Yu. Learning to predict middle-layer attention in MLLMs for visual token pruning. *arXiv preprint arXiv:2608.06411*, 2026b.

Rinyoichi Takezoe, Yaqian Li, Zihao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. LearnPruner: Rethinking attention-based token pruning in vision language models. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2604.23950.

Yao Tao, Yehui Tang, Yun Wang, Mingjian Zhu, Hailin Hu, and Yunhe Wang. Saliency-driven dynamic token pruning for large language models. *arXiv preprint arXiv:2504.04514*, 2025.

Yahong Wang, Juncheng Wu, Zhangkai Ni, Longzhen Yang, Yihang Liu, Chengmei Yang, Ying Wen, Lianghua He, Xianfeng Tang, Hui Liu, and Yuyin Zhou. When token pruning is worse than random: Understanding visual token information in VLLMs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2512.07580.

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

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. arXiv:2502.11501.

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.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22128–22136, 2025a. arXiv:2409.10197.

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

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research (TMLR)*, 2026. arXiv:2505.22654.

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. arXiv:2412.01818; an earlier version was titled "[CLS] Attention is All You Need for Training-Free Visual Token Pruning" (FasterVLM).

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)*, volume 38, 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.

Kai Zhao, Wubang Yuan, Yuchen Lin, Liting Ruan, Xiaofeng Lu, Deng-Ping Fan, Ming-Ming Cheng, and Dan Zeng. Attention debiasing for token pruning in vision language models. *arXiv preprint arXiv:2508.17807*, 2025.

Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. VisionSelector: End-to-end learnable visual token compression for efficient multimodal LLMs. In *ACM International Conference on Multimedia (ACM MM)*, 2026a. arXiv:2510.16598.

Qihui Zhu, Tao Zhang, Yuchen Wang, Zijian Wen, Mengjie Zhang, Shuangwu Chen, Xiaobin Tan, Jian Yang, Yang Liu, Zhenhua Dong, Xianzhi Yu, and Yinfei Pan. HAWK: Head importance-aware visual token pruning in multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b. arXiv:2604.07812.

A Protocol

Model and decoding. LLaVA-1.5-7B in fp16 with its CLIP ViT-L/14-336 tower (Radford et al., 2021) and MLP projector, the `vicuna_v1` template with the image before the question, pad preprocessing, greedy decoding with at most 128 new tokens (FastV's harness ran POPE, GQA and MME with at most 10, which no answer reached, and TextVQA with 128), and the suffix "Answer the question using a single word or phrase." This is the configuration of LLaVA's `model_vqa_loader`, which VisionZip and CDPruner evaluate with; our 576-token outputs reproduce the published MME and TextVQA numbers (Appendix J). Every timing is on one RTX 3090.

Answer recall. $a^\star$ and a_k are the decoded strings, stripped of surrounding whitespace and lowercased; recall is exact equality of the whole string. Nothing else is normalized, so a pruned answer that changes "Yes" into "Yes, there is" counts as a change.

Selection and ties inside a block. SHAPRUNE scores every token, replaces each score by the mean over its 3×3 block, and keeps the top k. All nine tokens of a block therefore carry the same score, and a budget that is not a multiple of nine takes part of a block: at $k = 64$ the selection is seven whole blocks plus one token of the eighth. The order is `np.lexsort((tiebreak, -score))`, where `tiebreak` is a fixed permutation of the 576 positions drawn from a generator seeded with $(\text{crc32}(u) \bmod 2^{31}, 3)$, where u is the row identifier (e.g. `pope_adversarial_1`). Ties are thus broken deterministically and identically for every score-based pruning method, and the same rule applies at 128 and 192 tokens; the harness that computed the Shapley reference seeded its own permutation from Python's salted hash of u; we therefore store that permutation with each of its records.

Matched random. For a row with identifier u let $s = \text{crc32}(u) \bmod 2^{31}$. One score vector $r \sim \mathcal{U}(0,1)^{576}$ is drawn from a generator seeded with $(s, 0, 7)$, ties are broken by a permutation seeded with $(s, 3)$, and the k highest-scoring tokens are kept in their original order. The same draw serves every budget, so the 64-token subset is nested in the 128- and 192-token subsets, and every method compared with matched random on that question sees the same subset. The tie-break permutation is shared by all score-based pruning methods; it matters for SHAPRUNE, whose block-averaged scores tie within a block.

Paired tests. Two methods are always compared on the same questions. For a binary per-question outcome, w counts the questions where only the first method succeeds and ℓ those where only the second does, and p is the p-value of the two-sided exact binomial test of w against $\text{Bin}(w+\ell, 1/2)$, i.e. the exact McNemar test. Recall is binary; for TextVQA, whose per-question accuracy is fractional, a question counts as discordant only when one method scores 1 and the other 0. As a sensitivity check we also tested TextVQA accuracy by an image-level sign-flip permutation test on the partial-credit differences (10,000 permutations, seed 0): of the 15 TextVQA accuracy cells against the baselines at layer 2 or before the LLM, 3 turn from ties into wins (VisionZip and CDPruner at 192 tokens, pure diversity at 128) and none is downgraded, and none of the three accuracy cells of the whole-answer head of §6 changes; because we saw the direction first, the registered verdicts stand. POPE's three categories share their 500 images and most yes-questions (9,000 rows, 5,170 distinct image–question pairs; score-based selectors give the same outcome on almost every copy, while matched random and the within-block tie-break are seeded per row, so their copies are different draws), so POPE tests keep the first row of each distinct pair in the order adversarial, popular, random. We also resample images with replacement (10,000 draws, generator seed 0) for a percentile interval on the difference in rates, computed on those same rows. The McNemar p treats questions as independent, which they are not — GQA's 12,578 questions share 398 images, POPE's 500, MME's 1,187 and TextVQA's 3,166 — so the interval, which resamples images, is the safeguard and a small p alone is never the evidence. For the two most quoted POPE cells at 64 tokens the two bases give SHAPRUNE after layer 8 a recall of 95.87% on the 9,000 rows and 96.32% on the distinct pairs (accuracy 86.56 and 89.25%) and CDPruner 94.69 and 94.99% (accuracy 87.84 and 89.19%). A difference is claimed only when $p < 0.01$ *and* the 99% interval excludes zero; everything else, including $0.01 \le p < 0.05$, is reported as a tie. Every difference between two methods quoted in the text is computed from the unrounded per-question rates and then rounded to one decimal, so it need not equal the difference of the rounded values printed in the tables. This rule was fixed before the de-duplicated tests were run, and it turned 11 of 163 stated differences into ties. Which rows the image bootstrap resamples was settled only after the first results of these tests had been seen, about 20 minutes later: we resample the de-duplicated rows, following our registered POPE definition, and did not change this afterwards; resampling all 9,000 rows instead could turn three borderline wins into ties (a post hoc check confirms that it does; none of the three is among the equal-kept-token comparisons). FastV runs in its own environment, so its *answer recall* is measured against the un-pruned answers that environment produces; those differ from ours on 6 of 9,000 POPE, 33 of 12,578 GQA, 9 of 2,374 MME and 23 of 5,000 TextVQA questions, and recomputing all twenty-four FastV cells against a single common reference leaves every verdict unchanged. SparseVLM and PyramidDrop are scored the same way; SparseVLM's unpruned answers differ from ours on 8, 23, 2 and 13 questions of POPE, GQA, MME and TextVQA, PyramidDrop's on none. CDPruner, although run in its own environment, is scored against our unpruned answers, which differ from its own on 11 of 9,000 POPE questions and raise its POPE recall by 0.03 points. Note what is and is not de-duplicated on POPE: *every rate printed in a table or in the text is over the 9,000 rows unless it is marked as on the distinct or de-duplicated pairs* (Tables 8, 9, 10, 2, 21 and 25 and the de-duplicated rates quoted in parentheses), while *every paired test, discordant count and interval is over the 5,170 distinct pairs*. The estimand of every POPE claim is thus the rate

over the 5,170 distinct pairs, each weighted equally; the 9,000-row rates are printed for comparability with published POPE numbers. The two differ because de-duplication drops repeated yes-questions and moves POPE's yes-rate from 50% to 30%, which lifts the weaker method more than the stronger one; wherever we quote a POPE gap of more than two points we therefore give both numbers. The smaller POPE gaps quoted in the main text, the placement gains of 0.4 to 2.1 points, move by less than 0.25 points when de-duplicated; Appendix F gives both values for them and for matched random's placement gains, which lie within 0.3 points of zero and move by up to 0.6. GQA repeats 8 image–question pairs among its 12,578 rows, too few to matter, and MME and TextVQA none, so we do not de-duplicate them and quote one number. Several cells pass the rule with p or an interval endpoint close to the threshold — among them the POPE cell of the first latency bracket (Appendix G), whose 99% interval starts at $+0.02$, and the two SHAPRUNE wins in Table 14 (intervals starting at $+0.04$ and $+0.02$); we flag none of them as borderline, and a reader who prefers a stricter line should treat those as ties.

Post hoc equivalence check. For the tie cells above we also ran, after seeing the data and with a margin chosen after the fact rather than registered, two one-sided tests (TOST, $\alpha = 0.05$ each side) against symmetric margins of 0.5, 1 and 2 points, read off the same image-cluster percentile bootstrap that produced the 99% intervals (10,000 resamples, seed 0), at its 90% level. This changes no verdict of §5.1–§5.3: a tie is a tie regardless of which margin is checked, and no cell is upgraded. At a margin of 1 point, SHAPRUNE after layer 8 and pure diversity after layer 8 (Table 8) are equivalent in 15 of the 24 recall and accuracy cells; the exception at 64 tokens is POPE recall (90% interval $[+0.19, +1.25]$, so a sub-point SHAPRUNE advantage is not ruled out), and several MME and TextVQA cells move outside 1 point while staying inside 2. The minimum effect these cells could have detected at 80% power ranges from 0.7 to 2.0 points, cluster-inflated by a factor of 0.95 to 1.37 over treating rows as independent — larger than 1 point at exactly the cells that fail the 1-point margin. We report this as a sensitivity analysis with a margin fixed after the results, not as a pre-registered equivalence claim.

Further post hoc checks. None of the following was registered, and none changes a verdict. Where SHAPRUNE and pure diversity after layer 8 tie, the two selections do not agree on which answers they keep: at 64 tokens, 4.8% of the 5,170 distinct POPE pairs and 15.5% of GQA questions keep the un-pruned answer under only one of the two — about eight times the 0.6%/2.0% rate at which two seeds of SHAPRUNE itself disagree on the same questions — so the tie reflects matched aggregate rates, not a shared selection. Taking whichever of the two keeps the answer on each question (a descriptive ceiling, not a method, since it uses the un-pruned answer to choose) would keep 98.4% of the distinct POPE pairs and 92.5% of GQA answers, 2.1 and 7.6 points above the better method alone. With each McNemar p replaced by an image-level sign-flip permutation p (10,000 sign flips, seed 0; every win at $p \leq 2 \times 10^{-4}$), all 69 recall wins of the Holm family of Appendix F (the 57 against pruners at layer 2 or before the LLM and the twelve against DivPrune) still survive Holm at 0.01. A paired difference-in-differences on the same POPE and GQA questions detects that the placement effect depends on the backbone: moving pure DPP or the trained head from before the LLM to after layer 8 raises recall by 1.4 to 4.9 points on LLaVA-1.5 in every cell but pure DPP's POPE cell at 64 tokens (0.6), against -0.1 to $+0.6$ on Qwen2.5-VL-7B, and the interaction's 99% image-bootstrap interval excludes zero in 5 of the 6 recall cells for each selector, all but POPE at 64 tokens (Appendix L). The 12 accuracy cells of pure diversity after layer 8 against SHAPRUNE, which belong to the registered comparison but are not printed in Table 8, are all ties (smallest $p = 0.06$, POPE at 128 tokens).

Benchmarks and scoring. MME (Fu et al., 2023) and TextVQA (Singh et al., 2019) come from LLaVA's evaluation files, POPE and GQA from the lmms-lab release of the same sets. *POPE*: the 9,000 test questions (adversarial, popular and random, 3,000 each). An answer maps to *no* if its first sentence, with commas removed, contains "No", "no" or "not" as a word, and to *yes* otherwise (LLaVA's `eval_pope`); F1 takes *yes* as positive within each category and the reported F1 is the mean over the three; accuracy and yes-rate are over all 9,000. *GQA*: testdev-balanced, 12,578 questions; the prediction with a trailing period removed and lower-cased must equal the answer. *MME*: LLaVA's `llava_mme.jsonl` (2,374 questions, two per image) on the images of the official MME release, scored by the official `calculation.py`; we report perception (P), cognition (C) and their sum, because VisionZip reports P+C and CDPruner P. Two quantities are called accuracy on MME: the official score just described, which combines per-question and per-image credit, and the plain

per-question correctness our paired tests use. We never read one off the other. *TextVQA*: LLaVA's `llava_textvqa_val_v051_ocr.jsonl` (the 5,000 validation questions with Rosetta OCR tokens in the prompt) on the official images, scored by LLaVA's `m4c_evaluator`. Run through our scoring code, the official answer files of LLaVA-1.5-13B reproduce its published MME perception (1531.3) and TextVQA accuracy (61.25).

Development subset and held-out questions. Every choice about the head — features, depth, loss, block size, checkpoint rule — was made on 1,500 POPE questions, 500 per category, drawn with a fixed seed (Table 16). Main-text benchmark results are on the full sets; numbers from the development subset, the reference questions or the held-out training pairs are labeled as such. On the other 7,500 POPE questions, which played no role in choosing the head (the placement after layer 8 was chosen after full-set results; §6), every POPE entry of Table 15 for matched random, VisionZip, CDPruner, pure DPP and SHAPRUNE before the LLM moves by at most 0.4 points, and SHAPRUNE before the LLM keeps a margin over matched random of +7.7 points of recall at 64 tokens (+4.9 deduplicated), against +7.2 (+4.0) on the full set. These questions are held out from method selection, not from its images (the development subset uses 486 of POPE's 500 images) and not wholly from its inputs: POPE repeats many image–question pairs across its splits, and 1,287 of the 7,500 held-out rows are identical to a development row, so this check shows stability to resampling rather than independence of the selection. Restricted to the 3,820 distinct POPE pairs that share no question with the development subset, SHAPRUNE after layer 8 wins all 21 recall cells against matched random, FastV at layers 2 and 8, VisionZip, CDPruner, pure diversity before the LLM and SparseVLM at the three budgets; against pure diversity at layer 8 two cells tie and one is a nominal win that, as registered, we do not use.

Training split. The 1,400 training pairs are VQAv2 (Goyal et al., 2017) train2014 questions on COCO images (Lin et al., 2014). A training pair is held out for checkpoint selection when $\mathrm{crc32}(\text{pair id}) \bmod 1000 < 100$, which selects 120 of the 1,400 pairs; the kept checkpoint is the one with the highest mean within-pair validation Pearson. Head seeds differ only in the order of the training pairs.

Registration history. We registered a matched random subset and pure diversity at the after-layer-8 placement and a head trained on the mean Shapley map, and canceled all three before any was evaluated; the first two were reinstated after an internal review, with their criteria recorded before the runs; the third was reinstated after a second internal review, also with its criteria recorded before the run (§5.4). A seed-1 head after layer 8, registered and canceled with them, was run later under its registered criteria, plus a tie criterion against pure diversity at layer 8 added before the run.

Registration status of SHAPRUNE-D. SHAPRUNE-D (the head's block scores as the relevance of the DPP after layer 8) was proposed after we saw that CDPruner's own selection moved to layer 8 keeps more GQA answers than SHAPRUNE; its criteria were recorded before its own runs: the four recall cells at 64 tokens against CDPruner@8 decide, and a quality claim needed no loss there, at least one win, and a win on fresh-image recall at 64 tokens. It won TextVQA and tied the rest, and the fresh-image confirmation failed, so no quality claim was made (Appendix F, Table 2). Its comparisons with pure DPP@8, SHAPRUNE@8 and SHAPRUNE-D before the LLM were registered as descriptive reports, not claims. The VizWiz test (decision cells: SHAPRUNE-D against pure DPP@8 and against CDPruner@8, *answer recall* at 64 tokens) was registered before the data were downloaded; both are ties. The equal-kept-token tally of SHAPRUNE-D (Appendix O) and its first-fresh-set comparisons other than CDPruner@8 at 64 tokens were computed afterwards, by the same scripts and rule as the registered tally of SHAPRUNE, whose win/tie/loss counts they reproduce exactly. The G5 test (10,000 further fresh VQAv2 images, disjoint from the first fresh set and from every training and pair-list image; decision cells E1, SHAPRUNE-D against CDPruner@8, and E2, SHAPRUNE-D against pure DPP@8, *answer recall* at 64 tokens after layer 8; the rule of Appendix A; accuracy reported, not used) was registered before its data were built, together with the sentence each verdict would put in the paper. Two framings were drafted before the G5 results were seen; which one is submitted was chosen after them. The overlap audit passed, so N is 10,000, and every run gate passed (all arms complete, the un-pruned answers reproduced, CDPruner@8 keeping the official token set on every question): SHAPRUNE-D keeps more answers than CDPruner@8 and than pure DPP@8 (§5.2).

B Answer recall and the bias shift

Definition. Let $a^\star(x, q)$ be the greedy answer of the un-pruned model to image x and question q, and $a_k(x, q)$ the answer after pruning to k tokens. Answer recall at budget k is $\text{Rec}_k = \Pr[a_k = a^\star]$, exact string match after lower-casing. Agreement with the uncompressed model has been measured as flips (Dutta et al., 2024) and loyalty (Wang et al., 2025); recall is such a measure. It is a fidelity measure: not whether the answer is right, only whether the selected tokens sufficed for the model to reproduce what it would have said. We report it beside accuracy and F1 throughout, and test every comparison on the same questions with an exact McNemar test on the discordant pairs. We adopted recall after our first experiment with the Shapley reference (Appendix D), in which accuracy did not separate it from VisionZip (16 against 20 discordant questions, $p = 0.62$) while recall did; the yes-rate analysis below is why we regard accuracy as the wrong instrument for ranking selectors, and we print accuracy beside recall in every cell so that a reader can apply either. POPE repeats most yes-questions across its three categories (9,000 rows, 5,170 distinct image–question pairs), so its tests count each distinct pair once; a difference is claimed only at $p < 0.01$ with a 99% image-level bootstrap interval excluding zero (Appendix A).

Why accuracy cannot rank selectors on POPE. Figure 3 decomposes what each 64-token method does to the un-pruned model's 9,000 answers. CDPruner changes 478 answers, breaking 200 correct ones and repairing 278 wrong ones; VisionZip changes 1,035 (804 broken, 231 repaired); matched random changes 1,152 (914 broken, 238 repaired). CDPruner's net accuracy is therefore *above* the un-pruned model's ($+0.9$ points) and its POPE F1 (87.5) exceeds the un-pruned 85.9, which no fidelity measure can explain. Part of the mechanism is the yes-rate: LLaVA-1.5 answers *yes* to 42.2% of POPE questions although the labels are balanced, CDPruner moves the rate to 46.6%, and VisionZip and random move it the other way, to 34.3% and 33.4%. Across twelve method–budget combinations — matched random, VisionZip, CDPruner and SHAPrune before the LLM, each at 192, 128 and 64 tokens — the deviation of the yes-rate from 50% correlates with F1 at Pearson -0.99; information loss and the yes-rate shift are confounded in that correlation, since losing evidence also pushes LLaVA toward *no*. Part of CDPruner's F1 advantage is recalibration and part is a genuine reduction of false-negative (omission) *no* answers, which we do not dispute. POPE itself reports the yes-rate (Li et al., 2023); what it cannot do is separate these two parts. What we dispute is that a column rewarding recalibration can rank selection signals.

What the recall failures are. Under a conservative normalization of surface forms (articles, punctuation, number words, regular plurals), 1.9 to 4.6% of the recall failures in each of the 36 method–budget cells on GQA and TextVQA are answers equivalent to the un-pruned one (at most 28 of 615, for SHAPrune after layer 8 on TextVQA at 192 tokens), so a failure is a change of content, not of form. On POPE and MME, whose answers are binary, a failure turns a correct answer wrong or a wrong one correct (Figure 3); 61 to 66% of TextVQA failures replace one wrong answer by another; and of the GQA failures at 64 tokens 35.9% turn a correct answer wrong for SHAPrune after layer 8, against 41.4% for CDPruner (descriptive, not tested).

Recall discriminates where accuracy does not. On the 200 GQA questions for which we computed the Shapley reference (Appendix D), its 64-token recall is 95.0% against 70.5% for a seed-0 random subset in the same harness — 50 questions recovered only by the reference against 1 only by random, $p = 5 \times 10^{-14}$ — while on the same questions accuracy is 57.0% against 52.0%, 21 wins to 11 losses, $p = 0.11$. One metric separates the two selectors decisively and the other does not separate them at this sample size. The reason is visible on the 83 questions the un-pruned model gets *wrong*: there the reference reproduces the wrong answer 94.0% of the time and is therefore 2.4% accurate, while the seed-0 random subset reproduces only 59.0% and is 14.5% accurate. Reproducing the model less is what repairs its errors. We state the premise this rests on: pruning is evaluated here as a cheap reproduction of the un-pruned model, not as a way to improve it.

A matched random baseline. A uniformly random subset of 64 tokens — the same seed per question for every method it is compared with, no CLS token, no merging — recovers 87.2% of answers on POPE and 74.1% on GQA (Table 3). Random subsets are known to be strong accuracy baselines (Wen et al., 2025a; Wang et al., 2026); scored by recall and matched per question they become the

floor of every paired test below. VisionZip's 64-token recall is 88.5% on POPE, 1.3 points above it over the 9,000 rows and 0.4 over the 5,170 de-duplicated pairs.

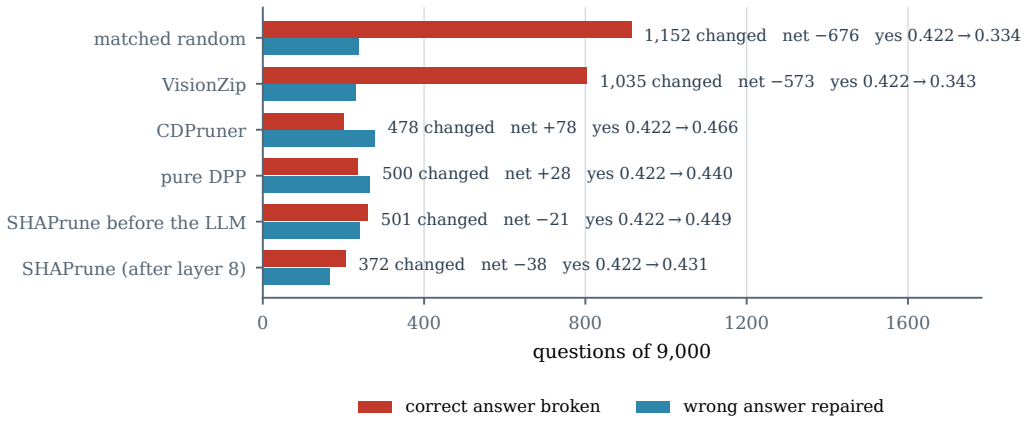

Figure 3: What a 64-token pruned model changes relative to the 576-token model on the 9,000 POPE questions: correct answers broken (red) and wrong answers repaired (blue), with the number of changed answers, the net change in correct answers (out of 9,000) and the yes-rate. That net change, annotated on each row, is the difference between the two bars. Their sum is the number of answers that changed at all — on POPE every textual change also flips correctness, so the two coincide exactly in all six rows — and that count is $9{,}000\,(1 - \textit{answer recall})$, the complement of recall rather than recall itself. The last row is top-k SHAPRUNE after layer 8; the row above it is the same scores applied before the LLM. The after-layer-8 row changes the fewest answers (372 of 9,000) and lands closest to the un-pruned yes-rate (0.431 against 0.422), which is the effect noted in the caption of Table 12: among the methods with the highest recall, the most faithful is the one POPE's accuracy column rewards least.

C Distance to the Shapley reference

The reference. Let the visual tokens be players and the value of a subset be the frozen model's log-probability of the first token of its own un-pruned answer when only that subset is kept, with the others removed — what pruning before the LLM does. The Shapley value of a token (Shapley, 1953) is its marginal contribution to that value averaged over all subsets, so it reflects the coalitions a token joins rather than its effect in isolation. Keeping the 64 tokens of highest Shapley value preserves LLaVA's answer on all 99 POPE questions and 95% of 200 GQA questions, against 92% and 84% for CDPruner, the strongest published pruner we ran on the same questions (a parameter-free diversity selector ties CDPruner on POPE before the LLM and keeps 95% and 84% after layer 8). The selection is post hoc — it needs the answer — and a reference rather than a bound, since many subsets preserve an answer. Its construction is in Appendix D.

Distance to the reference. Table 5 compares all methods with the Shapley selection on its 99 POPE and 200 GQA questions, each against its own un-pruned answers. At 64 tokens SHAPRUNE keeps 94.9% of POPE and 87.5% of GQA answers against 100 and 95.0% for the reference: it closes 50% and 67% of the gap between matched random before the LLM and the reference, pure diversity after layer 8 closes 50% and 52%, CDPruner 20% and 52% (descriptive; there SHAPRUNE and CDPruner do not differ significantly). The reference stays ahead on GQA (22 questions against 7, $p = 0.008$, 99% interval excluding zero) and on POPE loses no question to SHAPRUNE (5 against 0 over 97 distinct pairs, a tie). An answer-preserving subset is not unique, so the reference measures how often an answer-aware attribution finds *a* sufficient subset (Appendix D).

What the reference does and does not show. The reference plays the token-level game rather than the labels' block game (which, solved directly, keeps 100/99/99% of POPE answers), and token-

level single ablation, also answer-aware, ties it on all 99 POPE and 140 of its 200 GQA questions (Appendix D), so the reference shows headroom for answer-aware selection, not a property of Shapley values.

Table 5: Distance to the Shapley reference: *answer recall* on the questions where the Shapley selection was computed (99 POPE, 200 GQA), at 192 / 128 / 64 tokens, each method against its own un-pruned answers. The rows below the rule use the answer; they are references, not methods. Token-level: top-k tokens by ϕ; the block-averaged row averages the token-level ϕ on 3×3 blocks and is not the block game of the training labels; single ablation ranks tokens by $v(N)-v(N\setminus\{i\})$. Bold marks the best value in each column among the methods (the reference rows are excluded). A dash marks a selector we did not evaluate on that set: the block-averaged selection was run only in a POPE evaluation, so it has no GQA value. Not shown as rows: at 64 tokens SparseVLM, FastV moved to layer 8 and PyramidDrop keep 88.9, 86.9 and 82.8% of the POPE and 71.0, 70.0 and 52.5% of the GQA answers. ‡On GQA single ablation was computed for 140 of the 200 questions, on which the Shapley selection keeps 98.6 / 98.6 / 96.4%; on POPE it was run in the harness of the pruning methods, whose un-pruned answers match those of the Shapley harness on all 99 questions.

Method	POPE (99)	GQA (200)
Matched random	94.9 / 92.9 / 89.9	82.0 / 81.5 / 72.0
FastV (after layer 2)	93.9 / 92.9 / 85.9	83.5 / 77.0 / 64.0
VisionZip	96.0 / **97.0** / 90.9	86.5 / 83.0 / 76.5
CDPruner	91.9 / 90.9 / 91.9	85.5 / 85.0 / 84.0
Pure DPP	89.9 / 92.9 / 91.9	84.0 / 85.0 / 80.5
Matched random, after layer 8	97.0 / 92.9 / 92.9	87.5 / 84.0 / 79.0
Pure DPP, after layer 8	96.0 / 94.9 / **94.9**	90.0 / 89.5 / 84.0
SHAPRUNE (after layer 8)	**98.0** / 96.0 / **94.9**	**92.0** / **92.0** / **87.5**
Shapley selection, token-level (uses $a^\star$)	100 / 100 / 100	97.5 / 98.0 / 95.0
Token-level ϕ averaged on 3×3 blocks (uses $a^\star$)	99.0 / 99.0 / 99.0	–
Token-level single ablation (uses $a^\star$)‡	100 / 100 / 100	97.9 / 98.6 / 98.6‡

D The answer-conditioned Shapley reference

The game. For a fixed (x, q) with un-pruned answer $a^\star$, let $t^\star$ be its first generated token (the training labels of §3.2 skip the bare word-boundary token that precedes numerals; the two definitions coincide on the reference questions, none of whose 99 POPE and 200 GQA answers begins with a digit, and differ for 167 of the 1,400 training pairs) and define the value of a token subset $S \subseteq \{1, \ldots, D\}$ as $v(S) = \log p(t^\star \mid x_S, q)$, the log-probability the frozen model assigns to its own first answer token when only the visual tokens in S are kept — the others removed from the sequence, not masked, and positions re-indexed contiguously as in pre-LLM pruning. The Shapley value ϕ_i of token i is the standard average marginal contribution under this game, estimated by KernelSHAP (Lundberg & Lee, 2017) from 2,048 paired random coalitions per question, in the efficiency-constrained closed form of Covert & Lee (2021). Because v conditions on $a^\star$, ϕ is post hoc: computable only after the answer is known, at about five minutes per question on one GPU. We computed it for 99 POPE questions (33 per category) and 200 GQA questions, both drawn with a fixed seed, and call the top-k selector the Shapley reference of Appendix C. Note the asymmetry between the game and the metric: v scores one token while recall requires the whole answer string to match. The game is therefore a surrogate for the metric it serves — a cheap one, since scoring a coalition costs a single forward pass instead of a generation — and the 100% (POPE) and 95–98% (GQA) it reaches are empirical facts about that surrogate, not guarantees derived from it. Scoring the whole answer string instead, by teacher forcing, also costs one forward pass per coalition, on a sequence longer by the answer; we did not recompute the Shapley reference with it, but a head trained on whole-answer labels changes no TextVQA cell detectably (§6), and the token-level reliability figures of Appendix E use it.

Block games. Estimating ϕ for 576 players is expensive and, as we show below, unstable. We also define the game on 3×3 blocks of the 24×24 token grid, 64 players, and estimate block values from 1,024 coalition evaluations (16 per player, about as many as ViT-Shapley's amortized training

Table 6: The Shapley reference set: 99 POPE questions; cells are recall / accuracy / category-mean F1 at 192, 128 and 64 tokens. After layer 8 at 192 tokens SHAPRUNE reproduces 98% of the un-pruned answers, wrong ones included, so its accuracy and F1 there match the un-pruned model's (85.9 / 79.2 against 85.9 / 79.3) and are lower than every baseline's but FastV's (Appendix B). Paired against CDPruner on recall, the token-level reference wins 8–0, 9–0 and 8–0 ($p = 0.008$, 0.004, 0.008) and the block-averaged one 7–0, 8–0 and 7–0 ($p = 0.016$, 0.008, 0.016; a tie at our $p < 0.01$ threshold except at 128); neither loses a question. On accuracy the token-level selection is below CDPruner at 192 (0–8, $p = 0.008$) and the block-averaged one ties it (0–7, $p = 0.016$); both tie at 128 and 64. †Run in the harness that computed ϕ; all other rows in the harness of the pruning methods, with identical un-pruned answers. It is a reference, not a bound, and $n = 99$ cannot rank methods that tie at $n = 9{,}000$.

Method	192	128	64
Un-pruned (576)	100 / 85.9 / 79.3	100 / 85.9 / 79.3	100 / 85.9 / 79.3
Matched random	94.9 / 86.9 / 82.0	92.9 / 86.9 / 81.2	89.9 / 85.9 / 78.8
FastV (official, after layer 2)	93.9 / 79.8 / 65.5	92.9 / 80.8 / 66.4	85.9 / 75.8 / 51.3
VisionZip (official)	96.0 / 89.9 / 85.8	97.0 / 88.9 / 83.9	90.9 / 84.8 / 73.0
CDPruner (official)	91.9 / 93.9 / 91.6	90.9 / 90.9 / 87.4	91.9 / 87.9 / 83.5
SHAPRUNE after layer 8	98.0 / 85.9 / 79.2	96.0 / 87.9 / 83.2	94.9 / 86.9 / 81.9
Shapley reference, token-level, top-k (uses $a^\star$)†	100 / 85.9 / 79.3	100 / 85.9 / 79.3	100 / 85.9 / 79.3
Token-level ϕ averaged on 3×3 blocks (uses $a^\star$)	99.0 / 86.9 / 81.4	99.0 / 86.9 / 81.4	99.0 / 86.9 / 80.2

draws per patch: 32 subsets per image in each of 100 epochs (Covert et al., 2023)). Selecting the top blocks (seven, plus one token of an eighth) recovers the same POPE answers as token-level top-64 to within one question of 99, although the block selection preserves only 26% of the token identities. Those two figures come from the token-level ϕ averaged over 3×3 blocks (the row of Table 6 labeled accordingly); the 26% is the top-64 overlap, 0.261, between that block-averaged selection and the token-level one. The block game itself, solved directly on the same 99 questions from 1,024 coalition evaluations, keeps 100 / 99 / 99% of the un-pruned answers at 192 / 128 / 64 tokens (100 / 100 / 99% with a second, independent seed). Block attribution is also about twice as reproducible across independent estimates (overlap of the 64 selected tokens — seven blocks plus one token — 0.365 between the selections of these two 1,024-coalition block estimates on the 99 POPE questions, against 0.186 between two 2,048-coalition token estimates on 16 VQAv2 training pairs). The training labels of §3.2 use the block game.

The Shapley reference is numerically above every method on recall, and not above CDPruner on accuracy. Table 6 gives the 99-question comparison. On recall the reference is above every method at every budget and loses no question to CDPruner, but with seven to nine discordant questions the difference reaches our threshold at every budget for the token-level selection and only at 128 tokens for the block-averaged one. On accuracy it is below CDPruner at 192 tokens and ties at 128 and 64, for the reason of Appendix B: it reproduces the un-pruned model, wrong answers included, and CDPruner's accuracy advantage lives entirely on the 14 questions the un-pruned model got wrong. The GQA reference set shows the same direction at $n = 200$, where the reference beats CDPruner on recall at all three budgets (97.5 against 85.5, 98.0 against 85.0, 95.0 against 84.0; $p \leq 2 \times 10^{-4}$).

A second answer-aware selection does as well. Token-level single ablation, $v(N) - v(N \setminus \{i\})$, which also conditions on $a^\star$, keeps 100 / 100 / 100% of the 99 POPE answers and, on the 140 of the 200 GQA questions where it was computed, 97.9 / 98.6 / 98.6% against 98.6 / 98.6 / 96.4% for the Shapley selection on the same questions (no difference reaches our threshold; Table 5). On POPE it was run in the harness of the pruning methods, whose un-pruned answers match those of the Shapley harness on all 99 questions; its scores tie often there, so part of that selection is the seeded tie-break of Appendix A. On these questions the headroom of the reference is therefore that of answer-aware selection, not something only Shapley values provide.

The attribution itself is not reproducible, and the subsets are not unique. Two independent 2,048-coalition estimates of ϕ for the same question (16 VQAv2 training pairs, not the reference questions) overlap in their top-64 tokens by only 0.186 (chance 0.111); 16,384 coalitions per estimate raise this only to 0.33, on the 4 pairs we could afford at that budget (Figure 4, Appendix E). Yet both estimates recover the answer, and so does a random subset 87% of the time. Two objects are being

conflated. The Shapley value ϕ is unique, determined by the axioms from v; the set of 64 tokens that preserves $a^\star$ is not (Camburu et al., 2020). The reference's headroom is therefore not evidence that it identifies *the* important tokens, but that a signal aligned with the answer finds *an* answer-preserving subset more reliably than any answer-free proxy on GQA, and at least as reliably on POPE, where $n = 99$ cannot separate it from SHAPRUNE. The head is therefore trained to predict ϕ, which is unique but hard to observe, and evaluated on whether it preserves the answer, which many subsets do.

E RELIABILITY OF THE SHAPLEY REFERENCE

Figure 4 gives the agreement curves behind the top-64 overlaps quoted below and in Appendix D (0.186 at 2,048 coalitions, 0.33 at 16,384): how much two independent token-level estimates of the same question agree as the coalition budget grows. At small budgets the split-half figures quoted here reflect the estimator as much as the game: the two 64-coalition halves of a 128-coalition budget correlate at Pearson $+0.016$, and a perfectly additive synthetic game gives only $+0.06$ under the same procedure, because 576 players cannot be attributed from 64 coalitions each; at a budget of 1,024 the halves correlate at $+0.11$ (380 pairs), and two independent 2,048-coalition estimates share 0.186 of their top-64 tokens (chance 0.111; 16 pairs). These token-level figures, like the curves of Figure 4, score the summed log-probability of the whole answer rather than that of its first token (§3.1); scored on the first token they are $+0.013$ at 128 coalitions, $+0.11$ at 1,024 (on 400 pairs), and top-64 overlaps of 0.185 at 2,048 and 0.33 at 16,384 coalitions.

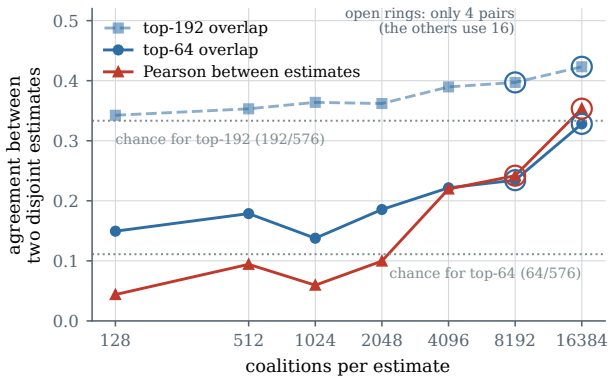

Figure 4: Agreement between two independent KernelSHAP estimates of the same question as a function of the number of coalitions per estimate. The game here is the *token-level* one with 576 players, not the 3×3 block game that produces the training labels (whose split-half reliability is $+0.52$). Each overlap curve has its own chance level, drawn as a dotted line: $192/576$ for top-192 and $64/576$ for top-64. The top-192 curve stays within 0.09 of its own chance. The two rightmost points of every curve rest on 4 image–question pairs (open rings); the others on 16. These are independent estimates on 16 pairs, whereas the $+0.11$ of §3.2 is between the two 512-coalition halves of one 1,024-coalition budget over the 380 pairs then cached, so the two numbers need not agree. The curves score the whole answer's log-probability rather than its first token's (first-token values in the text).

Block size (post hoc). On the same 1,024 coalitions the split-half reliability is $+0.111$ with single tokens as players, $+0.229$ with 2×2 blocks and $+0.519$ with 3×3 blocks. Coarser blocks cost selection resolution — a random subset at 64 tokens keeps 89.3% of the development-subset answers when drawn by token, 84.6% by 2×2 block and 80.4% by 3×3 block — yet the head trained on 2×2 blocks reaches at most 88.8% (best seed) against 94.1% for 3×3 blocks (development subset, before the LLM). We did not analyze small objects or text-reading questions separately.

F All ablation tables

Pruning after layer 8. Reusing the layer-0–7 keys and values the head has already computed and running only layers 8–31 on the kept tokens raises recall at every budget on POPE and GQA over the same scores applied before the LLM (on POPE by 1.7, 1.8 and 1.4 points at 192, 128 and 64 tokens, 1.5, 1.6 and 1.4 on the de-duplicated pairs), and is 8.8 ms cheaper at 64 tokens (Tables 12 and 17). The placement is FastV's (Chen et al., 2024), and the 8.8 ms is the recovery of the double computation a before-LLM design creates; attributing the recall gain requires separating the placement from the signal. The two paths select bit-identical token sets on the questions we checked, and repeating the after-layer-8 run with θ left uniform gives 67.0% recall at 64 tokens against 95.9% on POPE and 58.8% against 84.6% on GQA (discordant counts 1,050 against 123 over the distinct POPE pairs and 3,877 against 632 on GQA). With a below-random selector, moving the pruning point changes little (uniform head: 67.4% before the LLM, 68.5% after layer 8 on the same 1,500 development questions, $p = 0.03$, a tie). With the answer-free selectors of Table 3 the effect of the placement is not small: pure diversity after layer 8 gains 2.0, 2.1 and 0.4 POPE points (1.9, 2.0 and 0.6 de-duplicated), 4.8, 4.6 and 4.0 GQA points, 3.5, 4.0 and 4.4 MME points and 6.0, 6.4 and 5.3 TextVQA points at 192, 128 and 64 tokens (every cell a win except POPE at 64), and matched random gains 3.6, 3.6 and 2.8 GQA points and ties on POPE at every budget ($+0.3$, 0.0 and -0.3 points, $+0.6$, $+0.4$ and $+0.3$ de-duplicated); after layer 8 pure diversity ties SHAPrune in every POPE, GQA, MME and TextVQA cell and matched random loses every POPE and GQA cell (Table 12, Appendix J). On MME and TextVQA the head after layer 8 has the highest recall at every budget among the methods that prune before the LLM or at layer 2, except on TextVQA at 192 and 128 tokens, where FastV keeps as many answers at layer 2 and more at layer 8 (Appendix J).

Label-free readouts at layer 8. Against the random selector with SHAPrune's 3×3 block structure, SHAPrune's POPE margin at 64 tokens is $+9.5$ points on the de-duplicated pairs. FastV's score at layer 8 is its official code, which masks; applied once with the tokens dropped, at exactly our decoder FLOPs, it keeps 83.1 and 69.9%. *Debiased uniform θ* (registered before the run): from each of the 256 features of every visual token we subtract its mean at that token position over the images of 1,280 training pairs (no labels, no evaluation images), exclude the tokens that fall in LLaVA's square padding from the candidates, and average and select blocks as SHAPrune does. At 64 tokens it keeps 85.8% of POPE answers (89.96% of the 5,170 distinct pairs) and 75.2% of GQA answers, against 67.0 and 58.8% for uniform θ; SHAPrune keeps more on both (450 against 121 discordant pairs, $p = 2 \times 10^{-45}$, 99% interval $[+5.2, +7.5]$; 1,904 against 724 questions, $p = 3 \times 10^{-121}$, $[+8.0, +10.8]$), and matched random at layer 8 also keeps more than the debiased readout (POPE 91.3 against 90.0% on distinct pairs, $p = 0.003$, $[-2.5, -0.2]$; GQA 76.9 against 75.2%, $p = 10^{-4}$, $[-3.1, -0.3]$). The same debiasing was not applied to FastV's score, and no published debiasing method (Zhao et al., 2025) was run. *Stacked readout* (registered before the run): the same debiasing applied to uniform θ over layers 6–7 keeps 91.4% of POPE answers (93.35% of distinct pairs) and 78.3% of GQA answers; SHAPrune keeps more (263 against 109 discordant pairs, $p = 8 \times 10^{-16}$, 99% interval $[+2.0, +3.9]$; 1,453 against 659 questions, $p = 2 \times 10^{-68}$, $[+5.2, +7.5]$), while the stacked readout beats matched random (93.35 against 91.30% of distinct pairs, $p = 2 \times 10^{-6}$; GQA 78.3 against 76.9%, $p = 8 \times 10^{-4}$, $[+0.02, +2.77]$), debiased uniform θ and the undebiased layer-6–7 average ($p < 10^{-20}$ in all four cells). *HAWK-style head weights* (registered before the run): for each of the 256 heads of layers 0–7 we mask its attention to the visual keys and average, over 150 training pairs, the drop in the log-probability of the model's own first answer token; negative means are set to 0 and the weights normalized to sum to one, then read exactly as SHAPrune's θ. HAWK (Zhu et al., 2026b) itself uses the first layer, query and key without RoPE and a benchmark score. 88% of the weight falls on one layer-0 head (97% on layer 0). At 64 tokens it keeps 61.4% of POPE answers (74.89% of distinct pairs) and 53.3% of GQA answers; SHAPrune keeps more (1,228 against 120 discordant pairs, $[+20.1, +22.8]$; 4,466 against 530 questions, $[+29.5, +33.0]$; $p < 10^{-200}$), and the HAWK-style readout falls below matched random and uniform θ ($p < 10^{-28}$ in all four cells). Of the two undebiased label-free readouts of Figure 2, uniform θ over layers 6–7 keeps 12.1 POPE points fewer than the trained head at 64 tokens (7.6 de-duplicated), and FastV's score at layer 8 12.6 fewer (8.1 de-duplicated). Post hoc, matched random at layer 8 beats both on POPE and GQA ($p \leq 3\times10^{-8}$). The two readouts are statistically indistinguishable from each other (248 against 220 discordant POPE questions, $p = 0.21$; GQA interval $[-2.02, +1.01]$). The deeper placement helps FastV (POPE recall 78.7% at layer 2, 83.2% at layer 8), but it still trails SHAPrune on POPE and GQA at every budget. The shuffled-label head of §5.4 sits 21.4 POPE points above uni-

form θ over layers 0–7 (13.2 de-duplicated) and beats both undebiased label-free readouts on POPE (not the stacked readout) ($p \leq 1.5 \times 10^{-10}$) and on GQA by 4.8 to 8.4 points at the 99% level.

Table 7: Readouts of early-layer attention against the learned one: *answer recall* at 64 kept tokens (POPE: 9,000 rows, de-duplicated pairs in parentheses). *Condition*: *same* = same depth as SHAPRUNE after layer 8 and, for FastV, same decoder FLOPs in its token-dropping form (Table 18; the official run masks and computes all 576 positions); *shallower* = FastV's own layer 2; *nominal* = a layer-average budget below our decoder arithmetic. Every cell is a significant loss to SHAPRUNE (§4) except $^\circ$ (a tie). A dash: not run. *Diagnostic re-implementation (image boundary corrected, score applied once), not in the equal-token tally.

Readout of early-layer attention	Condition	POPE	GQA	MME	TextVQA
FastV (official, after layer 2)	shallower	78.7	67.8	85.5	76.4
FastV (official, moved to layer 8)	same	83.2 (88.2)	70.0	85.4	78.2°
FastV, corrected and applied once, layer 8*	same	83.1 (88.1)	69.9	85.7	75.0
SparseVLM (official)	nominal	88.0	72.1	84.5	69.2
PyramidDrop (official, 64 schedule)	nominal	71.0	57.6	73.5	61.2
Uniform θ, layers 0–7 (untrained), layer 8	same	67.0 (78.4)	58.8	–	–
Uniform θ, layers 6–7 (untrained), layer 8	same	83.8 (88.7)	69.5	–	–
SHAPRUNE (learned θ), layer 8	—	**95.9** (96.3)	**84.6**	**91.9**	**79.0**

Table 8: SHAPRUNE after layer 8 against pure diversity (CDPruner's DPP with constant relevance) moved to layer 8: *answer recall* on the full sets; POPE rates, like its tests, are on the 5,170 distinct pairs (so they differ from Table 12). w–ℓ: questions kept only by SHAPRUNE / only by pure diversity; the interval is the achieved 99% image-level bootstrap interval for SHAPRUNE minus pure diversity, in points. No cell passes the rule of Appendix A; these are the achieved intervals, not a pre-specified equivalence margin.

Benchmark	k	SHAPRUNE	Pure DPP	w–ℓ	p	99% interval
POPE	192	98.05	97.66	68–48	0.077	$[-0.15, +0.94]$
POPE	128	97.45	97.25	80–70	0.46	$[-0.41, +0.80]$
POPE	64	96.32	95.61	143–106	0.022	$[-0.12, +1.55]$
GQA	192	92.31	92.11	530–504	0.44	$[-0.54, +0.96]$
GQA	128	89.71	89.39	722–682	0.30	$[-0.55, +1.21]$
GQA	64	84.58	84.88	956–993	0.41	$[-1.34, +0.77]$
MME	192	96.00	95.87	51–48	0.84	$[-0.97, +1.22]$
MME	128	94.73	94.02	74–57	0.16	$[-0.55, +1.98]$
MME	64	91.87	92.17	96–103	0.67	$[-1.77, +1.22]$
TextVQA	192	87.70	87.48	264–253	0.66	$[-0.96, +1.43]$
TextVQA	128	84.16	84.82	294–327	0.20	$[-1.93, +0.66]$
TextVQA	64	79.00	79.56	369–397	0.33	$[-2.00, +0.87]$

Seed 1 against the baselines. With the seed-1 head, every one of the 16 registered win claims (§5.3; POPE and GQA recall at 64 tokens against matched random before the LLM and after layer 8, FastV at layers 2 and 8, VisionZip, CDPruner, pure diversity before the LLM and SHAPRUNE before the LLM, the last with the head of the same seed) passes the rule of Appendix A again, e.g. against CDPruner on POPE 156 against 86 distinct questions ($p = 8.0 \times 10^{-6}$, $[+0.56, +2.18]$) and against FastV at layer 8 on GQA 2,527 against 656; against pure diversity at layer 8 no difference is detected (POPE $p = 0.018$, $[-0.08, +1.55]$; GQA $p = 0.95$, $[-1.05, +1.00]$).

The head's scores as the relevance of the DPP after layer 8. SHAPRUNE-D runs a DPP selection after layer 8 with the head's block scores in place of CDPruner's CLIP question relevance. It was proposed after we saw that CDPruner@8 keeps more GQA answers than SHAPRUNE (Table 10); its criteria were recorded before its own runs: the four recall cells at 64 tokens are the decision cells, and

Table 9: Seed robustness of SHAPRUNE after layer 8: the seed-0 head of the paper against a second training seed (seed 1, which differs only in the order of the training pairs), full POPE and GQA sets; POPE rates and tests on the 5,170 distinct pairs. No cell passes the rule of Appendix A. At 64 tokens POPE recall and accuracy share their 14–15 discordant pairs (a changed yes/no answer flips both). MME and TextVQA were not run with seed 1.

Bench.	k	Metric	Seed 0	Seed 1	p	99% interval
POPE	192	recall	98.05	97.99	0.65	$[-0.15, +0.27]$
POPE	192	acc.	89.54	89.67	0.17	$[-0.38, +0.08]$
POPE	128	recall	97.45	97.52	0.57	$[-0.35, +0.19]$
POPE	128	acc.	89.56	89.44	0.34	$[-0.15, +0.39]$
POPE	64	recall	96.32	96.34	1	$[-0.29, +0.25]$
POPE	64	acc.	89.25	89.26	1	$[-0.29, +0.25]$
GQA	192	recall	92.31	92.34	0.83	$[-0.32, +0.26]$
GQA	192	acc.	61.31	61.31	1	$[-0.24, +0.25]$
GQA	128	recall	89.71	90.01	0.023	$[-0.60, +0.01]$
GQA	128	acc.	61.09	61.19	0.36	$[-0.35, +0.15]$
GQA	64	recall	84.58	84.85	0.045	$[-0.62, +0.08]$
GQA	64	acc.	60.14	60.18	0.76	$[-0.32, +0.22]$

a quality claim would need no loss there, at least one win, and a win on fresh-image recall at 64 tokens. Against CDPruner@8 it wins one of the four, TextVQA (80.82 against 79.38%, 323–251, $p = 0.003$, $[+0.18, +2.71]$), with no difference detected on POPE, GQA and MME; on the 5,000 fresh images of Appendix M no difference is detected (87.16 against 86.12%, 255–203, $p = 0.017$, $[-0.04, +2.14]$), so the TextVQA win is not confirmed and we make no quality claim; a tie means that no difference was detected, not that the two are equivalent. Over the 24 cells of Table 2 it wins 2 (TextVQA recall at 192 and 64 tokens) and ties 22. The other three columns are reported as descriptive results, not claims, for a variant chosen after seeing results: against pure DPP at layer 8 it wins 7 cells and ties 17 (at 64 tokens it wins POPE and GQA recall and ties MME and TextVQA recall), against SHAPRUNE at layer 8 it wins 8 and ties 16, and against the same selection before the LLM it wins 18 and ties 6; no column loses a cell. Its selection cost is the head's plus a DPP that grows with k, so it does not have SHAPRUNE's budget-independent selection; in a second timing run it reaches the first token at 64 tokens in 128.4 ms with the reference DPP and 122.8 ms optimized, 4.9 ms before and 2.4 ms after CDPruner@8's reference and optimized selections, and 4.3 and 5.1 ms after pure DPP@8's (means of paired differences, 99% intervals excluding zero; Table 4 and Appendix G). It was not part of the registered equal-kept-token tally; its 64-token row in Table 3 and its post hoc tally on the same axis (Appendix O) were added afterwards. The DPP is our own implementation, not bit-identical to CDPruner's official code; the CDPruner@8 run uses the same code with CDPruner's own relevance and was checked to keep the official token set on every question, so the two differ only in the relevance. As in CDPruner, relevance is min–max normalized per image, so the lowest-scoring block receives a relevance of about ϵ/range ($\epsilon = 10^{-6}$); since the greedy DPP gain of a token is bounded by its squared relevance, that block is effectively never kept: running the same selection with the head's scores on the 1,400 cached training images, it was selected in none of them at 64, 128 or 192 tokens (descriptive).

Out-of-distribution test on VizWiz (registered). Before downloading the data we registered a test on all 4,319 questions of the VizWiz VQA validation set (one question per image, photographed by blind users; outside the distribution of our labels and not in LLaVA-1.5's instruction data), with LLaVA-1.5's official VizWiz prompt for the un-pruned and every pruned run, *answer recall* at 64 tokens after layer 8, our test rule with image clusters, and two decision cells: SHAPRUNE-D against pure DPP and against CDPruner at layer 8. Neither shows a difference: SHAPRUNE-D keeps 79.88% of the un-pruned answers against 78.98% for pure DPP (230–191, $p = 0.064$, 99% interval $[-0.35, +2.13]$) and 79.67% for CDPruner@8 (213–204, $p = 0.70$, $[-1.02, +1.41]$). As registered references, SHAPRUNE keeps 77.12%: fewer than pure DPP (260–340, $p = 0.0012$, $[-3.29, -0.42]$), more than matched random at layer 8 (71.78%; 453–222, $p = 4 \times 10^{-19}$, $[+3.80, +6.88]$) and fewer than SHAPRUNE-D (137–256, $p = 2 \times 10^{-9}$, $[-3.98, -1.60]$).

Table 10: SHAPRUNE after layer 8 against CDPruner's own selection moved to layer 8 (its official selection code; the kept token sets equal the official ones in every row, and only the application point moves). Full sets; POPE rates and tests on the 5,170 distinct pairs; accuracy is per question. w–ℓ: questions only SHAPRUNE / only CDPruner gets right; the 99% image-level interval is SHAPRUNE minus CDPruner, in points. †: CDPruner at layer 8 keeps more answers (GQA recall at every budget); every other cell is a tie. Moved from before the LLM, CDPruner gains in 19 of these 24 cells and ties in 5 (recall in all twelve); at layer 8 it beats pure diversity in 5 of 24 cells (GQA recall at every budget, POPE recall at 192 and 64 tokens) and ties in 19; neither comparison loses a cell.

Bench.	k	Metric	SHAPRUNE	CDPruner@8	w–ℓ	p	99% interval
POPE	192	recall	98.05	98.20	32–40	0.41	$[-0.62, +0.29]$
POPE	192	acc.	89.54	89.61	34–38	0.72	$[-0.54, +0.38]$
POPE	128	recall	97.45	97.39	51–48	0.84	$[-0.46, +0.56]$
POPE	128	acc.	89.56	89.50	51–48	0.84	$[-0.44, +0.58]$
POPE	64	recall	96.32	96.46	83–90	0.65	$[-0.85, +0.54]$
POPE	64	acc.	89.25	89.34	84–89	0.76	$[-0.72, +0.52]$
GQA	192	recall	92.31	93.06	354–448	0.001	$[-1.38, -0.07]^{\dagger}$
GQA	192	acc.	61.31	61.40	238–250	0.62	$[-0.56, +0.36]$
GQA	128	recall	89.71	90.83	501–641	3.8×10^{-5}	$[-1.89, -0.33]^{\dagger}$
GQA	128	acc.	61.09	61.31	353–381	0.32	$[-0.88, +0.42]$
GQA	64	recall	84.58	86.93	672–967	3.3×10^{-13}	$[-3.22, -1.43]^{\dagger}$
GQA	64	acc.	60.14	60.68	517–584	0.047	$[-1.28, +0.19]$
MME	192	recall	96.00	96.12	33–36	0.81	$[-1.05, +0.80]$
MME	192	acc.	78.94	79.40	29–40	0.23	$[-1.35, +0.42]$
MME	128	recall	94.73	94.90	48–52	0.76	$[-1.26, +0.97]$
MME	128	acc.	79.15	78.98	52–48	0.76	$[-0.93, +1.26]$
MME	64	recall	91.87	92.38	81–93	0.4	$[-1.90, +0.93]$
MME	64	acc.	78.48	78.05	92–82	0.5	$[-0.97, +1.81]$
TextVQA	192	recall	87.70	87.32	239–220	0.4	$[-0.76, +1.54]$
TextVQA	192	acc.	58.28	57.76	77–65	0.36	$[-0.17, +1.20]$
TextVQA	128	recall	84.16	84.64	271–295	0.33	$[-1.74, +0.74]$
TextVQA	128	acc.	57.85	57.56	94–93	1	$[-0.48, +1.07]$
TextVQA	64	recall	79.00	79.38	348–367	0.5	$[-1.77, +1.05]$
TextVQA	64	acc.	56.08	56.35	127–142	0.39	$[-1.21, +0.67]$

Placement at equal kept tokens. The same learned scores applied before the LLM keep 94.4% of POPE and 80.7% of GQA answers at 64 tokens and, in composed time, 118.9 ms, tying CDPruner on POPE recall ($p = 0.94$) and trailing it on GQA (-1.5 points, $p = 4 \times 10^{-6}$), and keep fewer TextVQA answers than VisionZip at 128 and 64 tokens and than pure diversity at 64 (Appendix J); after layer 8 they gain 1.4 POPE and 3.9 GQA points and save 8.8 ms. Moving the pruning point is not a trade of time for answers: at 192 kept tokens the two placements cost 150.3 and 133.7 ms, and the deeper one keeps more answers on all four benchmarks while answering 16.6 ms sooner (Tables 12 and 22). The gain is not specific to our scores. At 64 tokens SHAPRUNE and pure diversity already tie before the LLM on POPE and GQA (94.4 and 94.4%, 80.7 and 80.8%; at 192 SHAPRUNE keeps more POPE answers, 96.2 against 95.5%, $p = 10^{-4}$), and the move to layer 8 adds similar amounts to each except on POPE at 64 tokens: POPE and GQA points, 1.4 and 3.9 for SHAPRUNE against 0.4 (a tie) and 4.0 for pure diversity at 64 tokens, 1.7 and 4.4 against 2.0 and 4.8 at 192.

How much of the margin is placement (post hoc). For SHAPRUNE after layer 8 and a diversity pruner B before the LLM at equal kept tokens, the recall margin splits exactly into a *signal* part, SHAPRUNE after layer 8 minus B moved to layer 8, and a *placement* part, B at layer 8 minus B before the LLM. Table 11 gives both parts and the placement share of the margin for pure DPP and CDPruner; parts and share are bootstrapped jointly over images (10,000 resamples, seed 0, 99% percentile intervals; POPE on the 5,170 distinct pairs), and every total margin's interval excludes zero. The share is a ratio of two differences, so its percentile interval is unstable wherever the margin is small; we report it only because every margin here is well away from zero, and even so it is wide where the margin is smallest (POPE at 64 tokens). The signal part, a plain difference with its own interval, is the more reliable quantity. Over pure DPP the placement accounts for 83.3 to 111.9% of

the margin in 11 of the 12 cells and for 46.4% on POPE at 64 tokens (interval $[-2.6, +118.4]$%), and the signal part is detected in no cell. Over CDPruner it accounts for 94.1 to 199.0% in all 12, and the signal part is negative on GQA at every budget, where CDPruner's own selection at layer 8 keeps more answers than SHAPRUNE (Table 10); a share above 100% means a negative signal part. The decomposition was not registered and changes no verdict; the registered statement is that a substantial part of the margin comes from the placement.

Table 11: Placement share of SHAPRUNE's recall margin at equal kept tokens (post hoc). Margin: SHAPRUNE after layer 8 minus the pruner before the LLM, in points (every 99% interval excludes zero); signal: SHAPRUNE after layer 8 minus the same pruner moved to layer 8, with its 99% interval; share: the placement part as a percentage of the margin, with its 99% interval. POPE rates on the 5,170 distinct pairs.

		Over pure DPP before the LLM			Over CDPruner before the LLM		
Bench.	k	Margin	Signal [99%]	Placement share [99%]	Margin	Signal [99%]	Placement share [99%]
POPE	192	$+2.32$	$+0.39$ $[-0.15, +0.94]$	83.3 [63.3, 107.6]	$+2.03$	-0.15 $[-0.62, +0.29]$	107.6 [86.8, 138.0]
POPE	128	$+2.22$	$+0.19$ $[-0.41, +0.80]$	91.3 [68.0, 123.5]	$+1.95$	$+0.06$ $[-0.46, +0.56]$	97.0 [74.8, 129.5]
POPE	64	$+1.33$	$+0.72$ $[-0.12, +1.55]$	46.4 $[-2.6, 118.4]$	$+1.33$	-0.14 $[-0.85, +0.54]$	110.1 [69.6, 230.0]
GQA	192	$+5.06$	$+0.21$ $[-0.54, +0.96]$	95.9 [82.4, 112.1]	$+4.10$	-0.75 $[-1.38, -0.07]$	118.2 [101.7, 139.8]
GQA	128	$+4.95$	$+0.32$ $[-0.55, +1.21]$	93.6 [78.0, 112.7]	$+4.00$	-1.11 $[-1.89, -0.33]$	127.8 [107.6, 154.7]
GQA	64	$+3.74$	-0.29 $[-1.34, +0.77]$	107.9 [83.2, 147.6]	$+2.37$	-2.35 $[-3.22, -1.43]$	199.0 [143.9, 324.6]
MME	192	$+3.66$	$+0.13$ $[-0.97, +1.22]$	96.6 [69.7, 133.3]	$+3.75$	-0.13 $[-1.05, +0.80]$	103.4 [80.2, 134.3]
MME	128	$+4.76$	$+0.72$ $[-0.55, +1.98]$	85.0 [62.5, 112.9]	$+4.84$	-0.17 $[-1.26, +0.97]$	103.5 [82.4, 131.5]
MME	64	$+4.09$	-0.29 $[-1.77, +1.22]$	107.2 [74.5, 163.5]	$+4.76$	-0.51 $[-1.90, +0.93]$	110.6 [83.1, 155.8]
TextVQA	192	$+6.24$	$+0.22$ $[-0.96, +1.43]$	96.5 [79.3, 117.6]	$+6.46$	$+0.38$ $[-0.76, +1.54]$	94.1 [78.2, 113.3]
TextVQA	128	$+5.72$	-0.66 $[-1.93, +0.66]$	111.5 [90.1, 142.2]	$+6.22$	-0.48 $[-1.74, +0.74]$	107.7 [89.4, 133.5]
TextVQA	64	$+4.70$	-0.56 $[-2.00, +0.87]$	111.9 [84.5, 158.2]	$+5.92$	-0.38 $[-1.77, +1.05]$	106.4 [84.6, 138.1]

Paired tests of SHAPRUNE against every baseline. Against matched random, FastV, VisionZip, CDPruner and pure diversity, at 192, 128 and 64 tokens on all four benchmarks, SHAPRUNE after layer 8 keeps significantly more answers in 57 of the 60 cells ($p \leq 4.04 \times 10^{-5}$) and ties the other three, all against FastV (on MME at 192 tokens, $p = 0.047$, and on TextVQA at 192 and 128, $p = 0.66$ and 0.80). On accuracy it is significantly higher in all 15 GQA cells, 9 of 15 MME cells, 7 of 15 POPE cells and 4 of 15 TextVQA cells, and tied in the rest; no baseline is significantly more accurate in any cell by our rule. A post hoc image-level sign-flip test on all 9,000 POPE rows instead of the 5,170 distinct pairs (10,000 flips, seed 0) finds CDPruner before the LLM more accurate at all three budgets (by 0.99, 0.89 and 1.29 points, $p = 0.001$, 0.004 and 7×10^{-4}, 99% intervals excluding zero), and SHAPRUNE more accurate than FastV at layer 8 at 192 tokens and than VisionZip at 128; every POPE recall verdict is unchanged. Adding the three same-placement comparisons of Table 12 — FastV at layer 8, uniform θ over layers 6–7, and the shuffled-label head, each on POPE and GQA at three budgets — gives 18 further recall cells, all won, and 18 accuracy cells of which 15 are won and 3 tied, again with no cell lost. FastV at layer 8 on MME and TextVQA adds six recall cells: won on MME at 128 and 64 tokens, tied on MME at 192 and on TextVQA at 64, and lost on TextVQA at 192 and 128 ($p = 9 \times 10^{-6}$ and 0.002); of its six accuracy cells one is won (MME at 64) and five are tied. SparseVLM and PyramidDrop add 16 recall cells, all won, which we count separately because their budget is a layer average (§4), and their accuracy cells are counted in Appendix K rather than here. Matched random and pure diversity moved to layer 8 add twelve POPE and GQA recall cells, the six against random won and the six against pure diversity tied, and pure diversity adds six MME and TextVQA cells (all six tied); none of their accuracy cells is lost. CDPruner's own selection moved to layer 8 adds twelve recall cells, nine tied and three lost (GQA at every budget; Table 10). DivPrune from its official code before the LLM adds twelve recall cells, all won ($p < 10^{-6}$), and twelve accuracy cells, won on GQA, MME and TextVQA at every budget and tied on POPE at every budget (over the 9,000 rows DivPrune is numerically more accurate at 192 and 128 tokens, 87.6 against 86.8% and 87.5 against 87.0%). The comparisons at the same number of kept tokens in §5.3 are the 60 cells above, FastV at layer 8's twelve, the eighteen of matched random and pure diversity at layer 8, CDPruner at layer 8's twelve and DivPrune's twelve; the supervision controls are not baselines and are not counted there. All tests use the rule of Appendix A.

DivPrune. We ran DivPrune from its official code (commit 799e2d9) at its official position, before the LLM on the projected visual tokens (greedy max–min diversity, $1 - \cos$ distance), on all four benchmarks at 192, 128 and 64 tokens (MME and TextVQA were added after the POPE and GQA runs, with the same code and checks). With the vision tower in fp32 and TF32 off, the protocol of every other baseline, its kept token sets equal those of the official evaluation path on all 50 checked questions at every budget; the official path's default setting (fp16 vision tower, TF32 on) selects different sets, so DivPrune's published numbers come from a different numerical setting than ours. Recorded before the runs, recall at 64 tokens on POPE and GQA decides two comparisons. SHAPRUNE after layer 8 keeps more answers than DivPrune on both (POPE 95.9 against 93.9%, 198–110 on the 5,170 distinct pairs, $p = 6\times10^{-7}$, 99% interval $[+0.80, +2.60]$; GQA 84.6 against 80.6%, 1,438–936, $p = 6 \times 10^{-25}$, $[+2.90, +5.10]$), and over the 24 recall and accuracy cells wins 21 and ties 3, POPE accuracy at every budget. No difference is detected between pure DPP before the LLM and DivPrune in either decision cell (POPE 94.4 against 93.9%, 93–74, $p = 0.16$, $[-0.29, +1.00]$; GQA 80.8 against 80.6%, 756–725, $p = 0.44$, $[-0.69, +1.13]$) or in 21 of the 24 cells; in the other three pure DPP is ahead (TextVQA recall and accuracy at 192 tokens, $p \leq 6.2 \times 10^{-4}$, and accuracy at 64, $p = 4.4 \times 10^{-4}$), and DivPrune is ahead in none; a tie means no difference was detected, not that the two are equivalent.

Multiple comparisons. The 57 recall wins among the 60 cells against pruners at layer 2 or before the LLM all survive a Holm correction at a family-wise error rate of 0.01 (largest $p = 4.04 \times 10^{-5}$), as do the twelve against DivPrune added to that family; of the 35 accuracy wins in those cells 31 survive it (not CDPruner on TextVQA and MME at 128 tokens, nor VisionZip and pure diversity on MME at 64; Benjamini–Hochberg keeps all 35), and SHAPRUNE's one win over the single-ablation head in Table 14 survives neither correction. At 64 tokens the POPE accuracy tie with CDPruner (§5.3) is 86.6 against 87.8% over the 9,000 rows and 89.25 against 89.19% on the 5,170 distinct pairs the test uses ($p = 0.90$).

Table 12: Pruning after layer 8 at every budget, with the controls of §5.4. The same SHAPRUNE scores, applied after layer 8 instead of before the LLM: the layer-0–7 keys and values computed for the head are reused and only layers 8–31 run on the kept tokens, with the original position ids. Full sets; recall / accuracy. Every recall cell of SHAPRUNE after layer 8 beats the corresponding pre-LLM cell in a paired exact test ($p < 10^{-7}$ throughout). On accuracy the verdict depends on the benchmark, and the POPE yes-rate is consistent with this: at 64 tokens it is 43.1% here against 44.9% for SHAPRUNE before the LLM and 46.6% for CDPruner, with the un-pruned model at 42.2%, so among these three the method that reproduces the un-pruned answers most faithfully is the one POPE's accuracy column rewards least (Appendix B). The indented rows repeat the run with only the map from the 256 attention features to one score changed: uniform weights over all eight layers, uniform weights over layers 6–7 alone, and the identical head trained on labels permuted across training pairs. Against three of the rows that share its placement — FastV at layer 8, uniform θ over layers 6–7 and the shuffled-label head — SHAPRUNE wins all eighteen recall cells and fifteen of eighteen accuracy cells by the rule of §4; the three exceptions, marked $^\circ$, are ties (POPE accuracy at 192 tokens against FastV, $p = 0.95$, and against uniform layers 6–7, $p = 0.45$; POPE accuracy at 128 tokens against uniform layers 6–7, $p = 0.024$). Moving the pruning point after layer 8 is worth $+1.4$ POPE points at 64 tokens and the supervision $+28.8$ ($+1.4$ and $+17.9$ on the de-duplicated rows the tests use; Appendix A). Bold marks the best pruned value in each column, on accuracy as on recall; the row name of our method is also bold. FastV's official implementation prunes inside the model too; the supervised head beats it in all twelve cells at layer 2 ($p < 10^{-5}$). Matched random and pure DPP appear before the LLM and, run by us with the same reuse of layers 0–7, after layer 8: after layer 8 matched random loses every recall cell to SHAPRUNE and pure DPP ties every one. The row 'Block random, after layer 8' draws the same per-question random score, averaged over the same 3×3 blocks before top-k as SHAPRUNE is; it tests whether the block averaging alone explains SHAPRUNE's margin over matched random (it does not: SHAPRUNE beats it on every recall cell, and it in turn loses every recall cell to matched random) and is a control outside the 114 axis cells.

Method	192	128	64
POPE 9,000 — recall / accuracy			
FastV (official, after layer 2)	90.6 / 81.4	86.5 / 77.7	78.7 / 70.8
FastV (official, after layer 8)	95.2 / 85.4°	90.8 / 82.5	83.2 / 75.0
Matched random	93.6 / 85.0	91.9 / 83.6	87.2 / 79.5
Pure DPP	95.5 / 87.6	94.9 / 87.6	94.4 / 87.3
Matched random, after layer 8	93.8 / 84.1	91.9 / 82.1	86.9 / 78.1
Block random, after layer 8	90.9 / 81.1	87.5 / 78.2	80.3 / 71.7
Pure DPP, after layer 8	97.5 / 87.3	97.0 / 87.3	94.9 / 86.1
CDPruner	95.6 / **87.8**	95.1 / **87.9**	94.7 / **87.8**
SHAPRUNE before the LLM	96.2 / 87.7	95.4 / 87.5	94.4 / 86.7
SHAPRUNE (after layer 8)	**97.9** / 86.8	**97.3** / 87.0	**95.9** / 86.6
same, uniform θ (layers 0–7)	88.6 / 79.8	80.2 / 71.8	67.0 / 59.3
same, uniform θ (layers 6–7)	95.6 / 85.4°	93.1 / 83.9°	83.8 / 75.3
same, labels shuffled across pairs	94.5 / 84.5	92.7 / 83.1	88.4 / 80.0
GQA 12,578 — recall / accuracy			
FastV (official, after layer 2)	82.0 / 57.8	76.9 / 55.8	67.8 / 51.5
FastV (official, after layer 8)	86.5 / 59.3	80.1 / 57.5	70.0 / 52.6
Matched random	83.3 / 59.1	79.9 / 58.0	74.1 / 55.3
Pure DPP	87.3 / 60.0	84.8 / 59.4	80.8 / 58.3
Matched random, after layer 8	86.9 / 60.4	83.5 / 58.8	76.9 / 56.8
Block random, after layer 8	83.3 / 58.5	78.5 / 56.4	70.9 / 53.0
Pure DPP, after layer 8	92.1 / 61.0	89.4 / 60.8	**84.9** / 60.0
CDPruner	88.2 / 60.3	85.7 / 59.8	82.2 / 58.6
SHAPRUNE before the LLM	87.9 / 60.2	85.2 / 60.0	80.7 / 58.4
SHAPRUNE (after layer 8)	**92.3** / **61.3**	**89.7** / **61.1**	84.6 / **60.1**
same, uniform θ (layers 0–7)	78.2 / 56.5	68.9 / 52.3	58.8 / 46.4
same, uniform θ (layers 6–7)	86.9 / 60.1	81.3 / 58.2	69.5 / 52.8
same, labels shuffled across pairs	86.9 / 59.9	82.8 / 58.7	76.4 / 55.9

Table 13: Label controls. The same 1,024 coalition evaluations per pair, re-solved three ways, each used to train the identical 256-weight head: the efficiency-constrained KernelSHAP solution (our labels), an unconstrained least-squares fit with a free intercept, and the difference of mean values with and without each block. The second was registered as Banzhaf values, but it does not estimate them: an unweighted linear fit returns Banzhaf values when coalitions are drawn uniformly (Hammer & Holzman, 1992), and these were drawn from the Shapley kernel, so the fit estimates Shapley values without the efficiency constraint, which the Pearson objective ignores; it correlates with our labels at $+0.92$. All three read the *same* coalitions and cost no forward pass beyond the 1.4M already paid, so this table compares estimators and not label budgets. The difference of means on these coalitions is, in expectation, also a positive affine function of the Shapley value, so all three rows estimate the same target; it is not a single-ablation label set either. The single-ablation label set — one forward pass per block ablation, 66 forward passes per pair against 1,026 — was run only at the deployed placement and is in Table 14. POPE development subset (1,500 questions), 64 tokens, pruning before the LLM; matched random is 89.3 / 81.5. All three beat matched random at $p < 10^{-5}$ and are mutually indistinguishable; Table 14 repeats the unconstrained and difference-of-means heads on the full sets after layer 8. Discordant counts are over the 1,350 distinct image–question pairs of the subset. The last row removes the supervision entirely; it is the worst row of the table, and because it never beats the Shapley head its final column reports its comparison against matched random instead.

Labels	Split-half	Corr. with ϕ	Recall	Acc	vs the Shapley head
Shapley (KernelSHAP, efficiency-constrained)	$+0.52$	1.00	94.1	87.0	—
Unconstrained least squares (registered as Banzhaf), seeds 0 / 1	$+0.67$	$+0.92$	94.0 / 94.4	86.9 / 86.9	7–8, 12–8 disc. (tie)
Difference of means (same coalitions, no extra forward passes)	$+0.55$	$+0.65$	94.2	87.0	16–15 disc. (tie)
None: uniform θ = mean attention	—	—	67.4	60.2	27–305 vs matched random (loss)

Table 14: Label sets at the deployed placement. The same 1,024 coalition evaluations per pair, re-solved by unconstrained least squares (registered as Banzhaf values, which it does not estimate; Table 13) and as the difference of mean values with and without each block, and a separately computed single-ablation set, $v(N)-v(N\setminus\{i\})$ from 66 forward passes per pair; each trains the identical 256-weight head, which is then run after layer 8 exactly as SHAPRUNE is. Full sets, recall / accuracy. No value is bolded because there is little to mark: against each of the three, every one of the twelve recall and accuracy cells is a tie with SHAPRUNE by the rule of §4 except the two marked. *: GQA accuracy at 64 tokens against the difference-of-means head, where SHAPRUNE wins by 0.4 points ($p = 0.0035$, 99% interval $[+0.04, +0.82]$). †: GQA recall at 192 tokens against the single-ablation head, where SHAPRUNE wins by 0.6 points ($p = 0.002$, $[+0.02, +1.19]$); the registered deciding cell, POPE recall at 64 tokens, is a tie. The first three label sets cost the same 1.4M forward passes (Table 13); the single-ablation set costs 92K.

Labels for the identical head, after layer 8	192	128	64
POPE 9,000 — recall / accuracy			
Shapley (KernelSHAP) — SHAPRUNE	97.9 / 86.8	97.3 / 87.0	95.9 / 86.6
Unconstrained least squares	98.0 / 86.9	97.2 / 86.8	95.8 / 86.5
Difference of means (same coalitions)	97.9 / 86.8	97.4 / 86.8	95.8 / 86.2
Single ablation (66 forward passes per pair)	97.8 / 86.8	97.2 / 87.0	96.3 / 86.6
GQA 12,578 — recall / accuracy			
Shapley (KernelSHAP) — SHAPRUNE	92.3† / 61.3	89.7 / 61.1	84.6 / 60.1*
Unconstrained least squares	92.0 / 61.4	89.4 / 61.1	84.4 / 60.1
Difference of means (same coalitions)	92.1 / 61.4	89.4 / 61.1	84.1 / 59.7
Single ablation (66 forward passes per pair)	91.7 / 61.3	89.4 / 60.9	84.3 / 59.9

Table 15: Full sets at every budget for the methods that prune before the LLM (after layer 8: Table 12; MME and TextVQA: Table 22). POPE 9,000: recall / accuracy / mean-F1; GQA 12,578: recall / accuracy.

Method	192	128	64
POPE 9,000 — recall / accuracy / F1			
Un-pruned (576)	100 / 87.0 / 85.9	100 / 87.0 / 85.9	100 / 87.0 / 85.9
Matched random	93.6 / 85.0 / 83.6	91.9 / 83.6 / 81.7	87.2 / 79.5 / 75.4
VisionZip	95.3 / 86.5 / 85.5	93.1 / 84.9 / 83.3	88.5 / 80.6 / 77.0
CDPruner	95.6 / 87.8 / 87.3	95.1 / 87.9 / 87.5	94.7 / 87.8 / 87.5
Pure DPP	95.5 / 87.6 / 87.1	94.9 / 87.6 / 87.1	94.4 / 87.3 / 86.5
DivPrune (official)	95.6 / 87.6 / 87.1	94.8 / 87.5 / 86.9	93.9 / 86.5 / 85.5
SHAPRUNE before the LLM	96.2 / 87.7 / 87.2	95.4 / 87.5 / 86.9	94.4 / 86.7 / 86.1
GQA 12,578 — recall / accuracy			
Un-pruned (576)	100 / 61.9	100 / 61.9	100 / 61.9
Matched random	83.3 / 59.1	79.9 / 58.0	74.1 / 55.3
VisionZip	85.6 / 59.3	81.3 / 57.7	75.6 / 55.1
CDPruner	88.2 / 60.3	85.7 / 59.8	82.2 / 58.6
Pure DPP	87.3 / 60.0	84.8 / 59.4	80.8 / 58.3
DivPrune (official)	87.1 / 60.1	84.5 / 59.3	80.6 / 57.7
SHAPRUNE before the LLM	87.9 / 60.2	85.2 / 60.0	80.7 / 58.4

Table 16: Development subset (1,500 POPE questions, 500 per category), on which every choice about the head was made. Recall / accuracy / mean-F1. The paper quotes this subset only where no full-set run exists (the label controls and the depth study). CLIP-CLS is the top-k by the vision encoder's class-token attention, the signal VisionZip builds on.

Method	192	128	64
Un-pruned (576)	100 / 87.6 / 86.1	100 / 87.6 / 86.1	100 / 87.6 / 86.1
Matched random	94.2 / 85.8 / 83.8	92.3 / 84.5 / 82.0	89.3 / 81.5 / 77.0
CLIP-CLS attention top-k	95.8 / 86.6 / 84.8	93.9 / 85.1 / 82.7	89.0 / 81.9 / 77.7
CDPruner	95.7 / 88.2 / 87.1	94.9 / 87.9 / 86.9	95.1 / 88.0 / 87.0
Pure DPP	95.3 / 87.7 / 86.6	94.8 / 88.0 / 87.0	94.8 / 88.3 / 87.1
Attention head, K=2	94.1 / 85.5 / 83.4	91.7 / 84.3 / 81.5	87.1 / 80.9 / 76.3
Attention head, K=4	94.4 / 86.1 / 84.2	92.1 / 84.5 / 81.8	87.9 / 81.1 / 76.8
Attention head, K=6	94.2 / 85.7 / 83.6	92.2 / 84.7 / 82.4	89.5 / 83.1 / 79.3
Attention head, K=8, uniform θ (untrained)	88.6 / 81.8 / 77.5	79.9 / 73.4 / 62.9	67.4 / 60.2 / 29.9
SHAPRUNE before the LLM (K=8)	96.5 / 88.0 / 86.8	95.1 / 87.5 / 86.4	94.1 / 87.0 / 85.7

G COST MEASUREMENT

Measured end-to-end latency and memory. The main text quotes measured times. On one RTX 3090 (fp16 LLM, batch size 1, GPU otherwise idle) we ran 300 real requests, the first 100 of a seeded shuffle of each of POPE, GQA and TextVQA after 10 warm-up requests each, through every pipeline and budget in a seeded random order, twice, and take the median of the two repeats per request. The primary quantity is the time to first token (image preprocessing to the first answer token); differences are paired per request, with 99% bootstrap intervals (10,000 resamples, seed 0) and are means of the paired differences (paired medians agree within 1.3 ms for every difference quoted). Every interval quoted in §5.5 excludes zero. Gates, registered before the measurement: the timed SHAPRUNE and pure-DPP-at-layer-8 paths return the answers of the evaluated runs on 150 POPE questions (all 150); the GPU-optimized DPP and CDPruner select exactly the reference token sets (all 300 requests, every budget); and a fused variant of SHAPRUNE (fused attention in layers 0–7, recomputing only the text rows it reads) had to select the same tokens as the eager path on 1,000 POPE questions in at least 99.5% of them. It did so in 97.7, 98.1 and 97.8% at 64, 128 and 192 tokens and failed, so SHAPRUNE is timed with its eager path, which materializes the attention of layers 0–7. Registered afterwards, a full-set test of the fused variant (ported into the evaluation path, selecting the same tokens as the timed fused path on 1,000 POPE questions at every budget) detects no difference from the eager path in *answer recall* at 64 tokens: POPE 96.36 against 96.32% on the distinct pairs (2–0, $p = 0.5$), GQA 84.55 against 84.58% (6–10, $p = 0.45$, 99% interval $[-0.12, +0.05]$). It reaches the first token in 115.3, 127.4 and 137.7 ms at 64, 128 and 192 tokens (eager 117.6, 129.4 and 140.1) with 152 to 154 MiB (eager 234). The un-pruned baseline runs the model's default fused attention (PyTorch SDPA). Measured answers agree with the evaluated runs on at least 99.67% of requests for every pipeline. Memory is the peak allocation during a request above the allocation just before it (weights resident), in MiB, taking the maximum over the two repeats and reporting the median over requests; it covers the vision encoder, the prefill, the selection and up to 128 decoded tokens on real prompts, and so differs from the 185.4 MiB below, which is the peak of the forward pass alone on a 611-position stand-in. Figure 5 plots the measured times against recall on GQA and TextVQA. The paired differences behind §5.5, at 64, 128 and 192 kept tokens: at layer 8 the reference DPP is 1.9, 7.9 and 14.0 ms slower than SHAPRUNE, CDPruner's reference selection 10.9, 24.2 and 37.3 ms slower, the optimized DPP 3.2, 1.8 and 0.6 ms faster, and the optimized CDPruner 0.9 ms faster at 64 tokens and 0.5 and 1.9 ms slower at 128 and 192; before the LLM, pure DPP answers 29.0, 19.5 and 8.3 ms sooner than SHAPRUNE, the optimized pure DPP 33.9, 28.4 and 21.3 ms sooner, CDPruner's optimized selection 31.5, 26.0 and 18.9 ms sooner, and its reference loop 20.5 and 4.4 ms sooner at 64 and 128 tokens but 14.1 ms later at 192. Materializing the layer-0–7 attention the head reads adds about 4.3 ms, $8 \times (5.23 - 4.69)$ ms (composed timings below). CDPruner's reference times use its own selection loop and CLIP text tower, its optimized times a GPU implementation that selects the same tokens (all 300 requests). SHAPRUNE-D was timed in a second run with the same protocol (gates: its answers equal the evaluated run's on all 150 POPE questions at every budget, and its optimized DPP selects the reference tokens on all 150; that path still synchronizes with the host for the CPU block mean), with SHAPRUNE's eager path, pure DPP and CDPruner at layer 8 and CDPruner before the LLM in both implementations, and the un-pruned model; the other GPU ran another job during the last 96 of its 300 requests. Its medians for the pruned pipelines shared with the first run were 2.8 to 6.6 ms higher (SHAPRUNE 3.8 to 4.4, pure DPP@8 4.1 to 5.0, CDPruner@8 4.6 to 6.6 ms), and the un-pruned model's 15.2 ms higher, so Table 4 keeps the first run's numbers and SHAPRUNE-D is compared within its own run. Paired there at 64, 128 and 192 tokens, the reference SHAPRUNE-D minus CDPruner@8's reference is -4.9 $[-5.9, -4.0]$, -12.9 $[-14.1, -11.7]$ and -20.2 $[-21.0, -19.4]$ ms, minus pure DPP@8's reference $+4.3$ $[+3.6, +4.9]$, $+4.2$ $[+3.4, +5.0]$ and $+3.9$ $[+3.1, +4.7]$ ms; the optimized SHAPRUNE-D minus the optimized CDPruner@8 is $+2.4$ $[+1.8, +3.4]$, $+1.8$ $[+1.2, +2.4]$ and $+2.4$ $[+1.7, +3.1]$ ms, minus the optimized pure DPP@8 $+5.1$ $[+4.4, +6.1]$, $+4.7$ $[+4.2, +5.1]$ and $+4.9$ $[+4.3, +5.5]$ ms; against SHAPRUNE the reference SHAPRUNE-D is 7.1, 12.4 and 18.5 ms slower and the optimized one 1.8, 2.2 and 4.0 ms (every interval excludes zero; paired medians within 1.1 ms of these means). Its measured answers agree with the evaluated run on 99.3 to 100% of requests.

Composed component timings. Before the end-to-end measurement the paper quoted times composed from component medians; they are kept below for reference, for the placement comparison and for the FLOPs–latency correlation of Table 18, and are labeled as composed wherever they appear (Tables 17 and 19). The main text uses the measured times above.

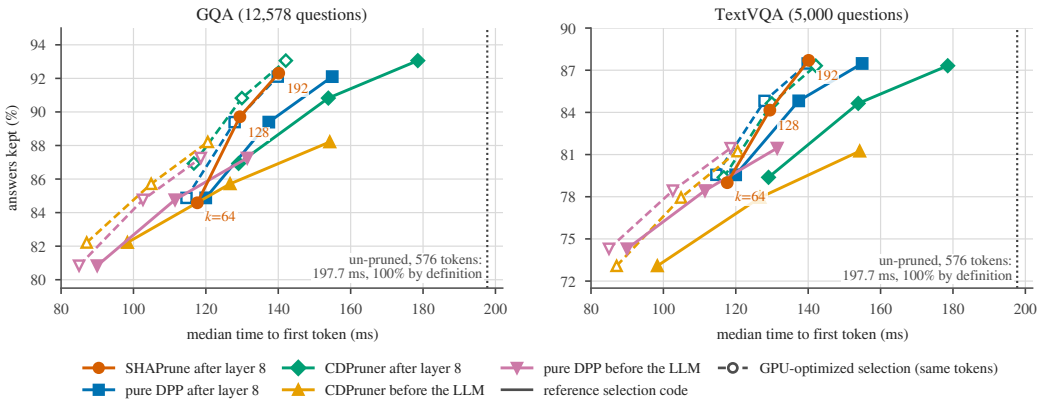

Figure 5: Measured time to first token against *answer recall*. Times are the medians of Table 4, over 300 mixed requests (100 each from POPE, GQA and TextVQA; one RTX 3090, batch size 1); recall is on the full GQA (12,578) and TextVQA (5,000) sets, so the two axes come from different samples. Each curve joins $k = 64$, 128 and 192 kept tokens, left to right. Solid lines use each method's reference selection code; dashed lines a GPU-optimized selection that returns the same tokens, so only the time moves. SHAPRUNE is shown with its eager pipeline only (the fused variant failed its token-set gate; its answers and times: Appendix G text). The un-pruned model takes 197.7 ms. VisionZip, matched random, FastV, SparseVLM, PyramidDrop, and SHAPRUNE before the LLM were not timed end to end, and SHAPRUNE-D is not drawn (its times, from a second run: Table 4).

Times are medians on one RTX 3090 in fp16 for a one-token answer, composed from three measurements: the CLIP encoder, projector and every selection cost over 30 different POPE images; the LLM prefill over 100 runs of one synthetic image, without a key–value cache, over $35 + k + 26$ positions (637 un-pruned, the most common POPE length); and the saving from reusing layers 0–7 over 30 runs on random hidden states of a 611-position stand-in (35 text and 576 visual positions; it keeps the 35 text positions, the first k visual ones and the last 11 as a stand-in for the question). Measured again on the POPE geometry, 638 positions of which $62 + k$ are kept, the saving is 9.1, 13.5 and 17.4 ms at 64, 128 and 192 tokens, 0.3 to 0.9 ms more than on the stand-in, so our after-layer-8 times are conservative by that much. The remaining 5.4 ms is what an end-to-end un-pruned `generate()` call with one new token costs beyond its prefill and the CLIP time of that run (33.8 ms); no separate decode forward runs for a one-token answer. The CLIP encoder and projector take 34.4 ms in the selection-cost runs, the value every total uses, so every absolute time, un-pruned included, is about 0.5 ms above what a single run would give; comparisons are unaffected. Selection cost is what is paid before pruning: 41.8 ms for SHAPRUNE, which runs the first 8 LLM layers with attention weights materialized (10.4, 21.0 and 31.3 ms for $K = 2, 4, 6$), 12.0 ms for CDPruner's selection operations as we re-measured them (CLIP text tower, cosine similarities, DPP) and 7.0 ms for the greedy DPP alone, both at 64 kept tokens. The greedy DPP takes one step per kept token, so we timed both again at every budget over 100 images: within 4% of these values at 64 tokens (7.3 ms for the DPP and 12.3 ms for CDPruner in that run; we quote the first run's 64-token values), they take 19.9 and 25.9 ms (CDPruner) and 14.2 and 20.6 ms (DPP alone) at 128 and 192. CDPruner's own loop, which selected the same tokens on the image we compared at each budget, is slower (35.0 and 49.4 ms); we time our port, so the composed CDPruner time at 192 tokens is 23.5 ms below its own loop's. The LLM prefill takes 150.0 ms with all 576 visual tokens and 68.8, 54.5 and 37.4 ms with 192, 128 and 64. End to end at 64 tokens this gives 77.1 ms for matched random and VisionZip (whose own selection cost we did not time and count as zero), 84.2 ms for pure DPP, 89.1 ms for CDPruner, 118.9 ms for SHAPRUNE before the LLM and 110.1 ms for SHAPRUNE after layer 8, against 189.7 ms un-pruned; at 128 and 192 tokens pure DPP takes 108.5 and 129.1 ms and CDPruner 114.1 and 134.4 ms. Pruning after layer 8 reuses the layer-0–7 keys and values and runs only layers 8–31 on the kept tokens; a first measurement on the same sequence gave 28.1 ms instead of 36.7 ms for all 32 layers, a saving of 8.6 ms. A later run measured the same quantity at all three budgets (Table 17) and gives 8.8 ms at 64 tokens, which is what we subtract. We quote this newer set (110.1 / 123.4 / 133.7 ms) everywhere, because it is one run under one protocol and therefore internally consistent; the earlier single-budget measurement gave 110.3 ms for the 64-token case, 0.2 ms away.

Table 17: Pruning after layer 8, composed times at every budget (component medians of 30 runs, one RTX 3090, fp16), for a one-token answer: the un-pruned 189.7 ms of the speed-up row is this older composed measurement, and the end-to-end measurement of Table 4 gives 197.7 ms. *Saving* is the difference between re-running all 32 layers on the kept tokens and running only layers 8–31 over the reused layer-0–7 keys and values. Bold marks the line this paper quotes. *End to end* subtracts the saving from the corresponding SHAPRUNE before-the-LLM time; the 64-token value is 0.2 ms below the earlier single-budget measurement of 110.3 ms because the saving was re-measured (8.80 against 8.60 ms).

Kept tokens	64	128	192
SHAPRUNE before the LLM (ms)	118.9	136.0	150.3
Saving from reusing layers 0–7 (ms)	8.80	12.63	16.64
SHAPRUNE after layer 8, end to end (ms)	**110.1**	**123.4**	**133.7**
Speed-up over the un-pruned model (189.7 ms)	1.72×	1.54×	1.42×
POPE *answer recall* (%)	95.9	97.3	97.9

Components are rounded medians, so totals can differ from their sum by 0.1 ms. Our selection recomputes the attention of the first 8 layers eagerly, at 5.23 ms per layer, head included, against 4.69 ms per layer for a fused prefill that also includes the final norm and LM head (4.82 ms in the run that timed the selection), so an implementation that reads the weights out of the prefill would be roughly 3 to 5 ms cheaper; we report what we ran. That difference is not arithmetic: the matrix products are identical, materializing the softmax adds 0.5 GFLOPs (0.006% of the un-pruned forward pass) and the head itself 0.0043 GFLOPs. The cost is memory traffic, not FLOPs.

Pure DPP at our layer, at matched compute. Pure DPP moved to layer 8 at 64 tokens, in place of SHAPRUNE in Table 19, keeps 94.9, 84.9, 92.2 and 79.6% of POPE, GQA, MME and TextVQA answers and stands almost exactly as SHAPRUNE does against the same rows: it ties VisionZip, CDPruner, SparseVLM and pure DPP before the LLM, all at 192 tokens, on POPE and MME, loses TextVQA to all four and GQA to CDPruner and to pure DPP at 192, and beats matched random on GQA, MME and TextVQA. Two of its 20 verdicts differ from SHAPRUNE's: against matched random on POPE it ties ($p = 0.052$), and against SparseVLM on GQA it ties at the edge of our rule ($p = 0.003$, interval $[-2.05, 0.00]$). The matched-compute losses therefore come with pruning 64 tokens after layer 8, with either selector, rather than with the supervision; at matched compute the same selector keeps more answers with 192 tokens before the LLM on GQA and TextVQA.

At matched compute SHAPRUNE at 64 tokens also loses TextVQA accuracy to CDPruner, pure DPP and SparseVLM at 192 (56.1 against 57.3, 57.4 and 57.7).

SHAPRUNE-D at matched compute (post hoc). With the matching and rule of Table 19, DivPrune added, SHAPRUNE-D after layer 8 at 64 tokens was compared post hoc with each pruner at 192 tokens before the LLM (Table 20); its DPP's selection time, which grows with k, is not part of the match. It loses one of the eight recall and accuracy cells to CDPruner, GQA recall (86.60 against 88.21%, -1.61 points, 99% interval $[-2.48, -0.71]$), and ties the other seven; it wins POPE recall against pure DPP and ties its other seven cells; against SparseVLM it loses TextVQA recall (80.82 against 84.34%, $[-5.04, -2.00]$), wins POPE recall and GQA accuracy and ties the rest; and it wins all eight cells against matched random and three each against VisionZip and DivPrune, tying the rest. Where SHAPRUNE at matched compute loses GQA recall to CDPruner, pure DPP and SparseVLM and TextVQA recall to these three and VisionZip, SHAPRUNE-D thus keeps one GQA loss (to CDPruner) and one TextVQA loss (to SparseVLM).

Latency brackets (composed times). At a matched number of kept tokens CDPruner is faster than SHAPRUNE after layer 8 at 64 and 128 tokens (89.1 and 114.1 against 110.1 and 123.4 ms) and 0.7 ms slower at 192 (134.4 against 133.7 ms), so we also pair each of our budgets with CDPruner's next one, a comparison we chose after the latencies and the matched-FLOPs verdicts (Table 19) were known. Against CDPruner at 128 tokens, which answers 4.0 ms after we do at 64, we keep narrowly more answers on POPE (95.9 against 95.1, $p = 0.005$, 99% interval $[+0.02, +1.66]$) and MME (91.9 against 89.9, $p = 0.003$), fewer on GQA (84.6 against 85.7, $p = 0.002$) and as many on TextVQA

Table 18: Decoder prefill FLOPs and token-layers. FLOPs count the language model only: batch 1, the prefill plus the LM head for the first answer token, $2mpq$ per $(m\times p)(p\times q)$ product, per layer, for n positions, $8nd^2$ for Q, K, V, O, $4n^2d$ for attention and $6ndd_{\text{ff}}$ for the SwiGLU MLP, with $d = 4096$, $d_{\text{ff}} = 11008$ and 32 layers read from the checkpoint config. Normalization, RoPE, softmax and the activation are excluded, as are the CLIP tower and projector, which every method pays alike. The sequence is 638 positions (35 text, 576 visual, 27 text), that of the first POPE question; POPE questions span 637–640 positions (mean 637.7), which moves no row by more than 0.03 TFLOPs and no ratio by more than 0.25 points. Our code, like `transformers`, applies the LM head at every prefill position; counted that way, after layer 8 at 64 tokens is 0.23% cheaper than before the LLM at 192 rather than 0.78% more expensive. Counting projections and attention as multiply–accumulates would lower every row by 16.7–17.4%, and counting the MLP that way too would halve it (SparseVLM's formula, which counts this way, also takes the FFN width equal to d, lower still); we do not compare these absolute numbers with published ones, whose sequence assumptions differ, only the ratios within this table. The rows before the LLM (†) carry CDPruner's recall, whose selection runs outside the decoder. SHAPRUNE before the LLM is not such a row: its scoring pass, 8 layers over all 638 positions, adds 2.119 TFLOPs, for 5.443, 4.599 and 3.760 TFLOPs at 192, 128 and 64 tokens (64.2, 54.3 and 44.4%), more than after layer 8 at every budget (its own POPE recall before the LLM: 96.2, 95.4, 94.4). FastV rows are the token-dropping variant (see "Which FastV we ran, and which we count" below; the masking variant we ran costs the un-pruned 8.477); the SparseVLM rows count its published schedules and the PyramidDrop row the 64-token schedule its authors recommend, schedules only (PyramidDrop's ranking pass, Q, K and V of the next layer over 638, 171 and 82 positions, would add 0.090 TFLOPs, for 1.726), while the recall beside them is our run of their official code (Appendix K). Note what the last four rows show: at the same nominal budget these two methods spend far less decoder arithmetic than SHAPRUNE does, which is why Table 19 compares at matched compute as well. FLOPs are not end-to-end time: over the seven settings with a composed latency they track decoder prefill latency closely (Spearman $+0.96$, or $+1.00$ with our layer-0–7 pass timed as in the end-to-end totals) and end-to-end latency only loosely ($+0.36$). Our selection is the layer-0–7 pass, which the FLOPs already count. The end-to-end $+0.36$ uses SHAPRUNE's before-the-LLM latencies, which include a 41.8 ms scoring pass the FLOPs do not count; with CDPruner's latencies, whose selection is timed at each budget, it is $+0.71$. Bold: the after-layer-8 placement of SHAPRUNE and SHAPRUNE-D (POPE recall of SHAPRUNE).

Setting	Token-layers (visual)	TFLOPs	of un-pruned	POPE *answer recall*
Un-pruned, 576 tokens	18,432	8.477	100.0%	100
Before the LLM†, $k = 192$	6,144	3.324	39.2%	95.6
Before the LLM†, $k = 128$	4,096	2.480	29.3%	95.1
Before the LLM†, $k = 64$	2,048	1.641	19.4%	94.7
After layer 8, $k = 192$	9,216	4.612	54.4%	**97.9**
After layer 8, $k = 128$	7,680	3.979	46.9%	**97.3**
After layer 8, $k = 64$	6,144	3.350	39.5%	**95.9**
FastV K=2, $k = 64$	3,072	2.068	24.4%	78.7
FastV K=8, $k = 64$	6,144	3.350	39.5%	83.2
SparseVLM, $k = 192$	6,608	3.525	41.6%	95.2
SparseVLM, $k = 128$	4,653	2.723	32.1%	95.0
SparseVLM, $k = 64$	2,632	1.893	22.3%	88.0
PyramidDrop, 64 schedule	2,014	1.637	19.3%	71.0

(a tie), with all four accuracies tied. Against CDPruner at 192 tokens, which answers 11.0 ms after we do at 128, we keep more on all four ($p \leq 3 \times 10^{-5}$) and are more accurate on GQA (61.1 against 60.3, $p = 0.004$), tying the rest. With the measured times of Table 4 the brackets hold for CDPruner's reference selection (126.7 and 154.2 ms against our 117.6 and 129.4) but not for its optimized one (104.9 and 120.5 ms).

Which FastV we ran, and which we count. FastV ships two variants. The accuracy path its configuration enables by default masks the dropped positions rather than removing them, and that is the variant we ran, so our FastV recall and accuracy reflect its official default behavior. Because

Table 19: Matched compute rather than matched tokens (times in the second column are measured medians of Table 4; those in this caption are composed): SHAPRUNE after layer 8 at 64 kept tokens against the methods we ran at 192, which cost the same 6,144 visual token-layers and the same decoder FLOPs to within 0.8% (Table 18). The second column says how well each row is matched, because FLOPs are not the whole cost: matched random selects for free and VisionZip's selection, which we did not time, is counted as zero, so both answer at about 108.5 ms against our 110.1 (composed); CDPruner and pure DPP match on FLOPs but pay 25.9 and 20.6 ms of selection at 192 tokens, putting them at 134.4 and 129.1 ms (composed); SparseVLM is matched only approximately, spending 5% more decoder FLOPs than we do (6,608 token-layers, 3.525 against 3.350 TFLOPs), and was not timed here. FastV after layer 2 at 192 is not a row: as run it masks (576 tokens per layer), and its token-dropping form would cost 8.8% more FLOPs than ours; it keeps 90.6, 82.0, 94.9 and 87.9% of POPE, GQA, MME and TextVQA answers, so SHAPRUNE keeps more on POPE and GQA and fewer on MME and TextVQA. POPE rates are over the 9,000 rows, its tests over the 5,170 distinct pairs. Cells give the other method's *answer recall* with the verdict for SHAPRUNE in parentheses, by the rule of §4. VisionZip on GQA is a tie because the 99% interval $[-2.12, +0.11]$ contains zero, although $p = 0.006$. SHAPRUNE before the LLM is deliberately absent: it pays a scoring pass over all 576 tokens through 8 layers that these FLOPs do not count (41.8 ms), so it is not compute-matched with our row at all; the comparison with it is at equal kept tokens, where the after-layer-8 placement keeps more answers on all four benchmarks and is 16.6 ms faster at 192 (Tables 12, 22 and 17).

Method at $k = 192$	Matched on	POPE	GQA	MME	TextVQA
SHAPRUNE after layer 8, $k = 64$	117.6 ms (measured)	95.9	84.6	91.9	79.0
Matched random	FLOPs and time	93.6 (win)	83.3 (win)	89.6 (win)	71.9 (win)
VisionZip	FLOPs and time	95.3 (tie)	85.6 (tie)	92.5 (tie)	82.0 (loss)
CDPruner	FLOPs; 36.6 (reference) or 2.9 (optimized) ms slower, measured medians	95.6 (tie)	88.2 (loss)	92.2 (tie)	81.2 (loss)
Pure DPP	FLOPs; 13.8 (reference) or 1.0 (optimized) ms slower, measured medians	95.5 (tie)	87.3 (loss)	92.3 (tie)	81.5 (loss)
SparseVLM	+5% FLOPs; not timed	95.2 (tie)	85.9 (loss)	93.3 (tie)	84.3 (loss)

Table 20: SHAPRUNE-D after layer 8 at 64 kept tokens against pruners at 192 tokens before the LLM, matched on visual token-layers (post hoc). Cells: the comparator's recall / accuracy, then the verdict for SHAPRUNE-D on recall / accuracy (W win, T tie, L loss) by the rule of §4. POPE rates are over the 9,000 rows, its tests over the 5,170 distinct pairs, so a POPE rate difference and its verdict can differ in sign.

Method	POPE	GQA	MME	TextVQA
SHAPRUNE-D after layer 8, $k = 64$	96.40 / 87.13	86.60 / 60.52	93.30 / 78.90	80.82 / 57.01
Matched random, $k = 192$	93.58 / 85.02 (W / W)	83.33 / 59.14 (W / W)	89.55 / 76.20 (W / W)	71.90 / 52.83 (W / W)
VisionZip, $k = 192$	95.28 / 86.48 (W / T)	85.59 / 59.25 (W / W)	92.50 / 77.80 (T / T)	82.02 / 57.24 (T / T)
CDPruner, $k = 192$	95.61 / 87.77 (T / T)	88.21 / 60.34 (L / T)	92.25 / 77.97 (T / T)	81.24 / 57.31 (T / T)
Pure DPP, $k = 192$	95.46 / 87.61 (W / T)	87.26 / 60.05 (T / T)	92.33 / 77.55 (T / T)	81.46 / 57.36 (T / T)
SparseVLM, $k = 192$	95.22 / 86.64 (W / T)	85.93 / 59.49 (T / W)	93.26 / 77.76 (T / T)	84.34 / 57.73 (L / T)
DivPrune, $k = 192$	95.61 / 87.61 (W / T)	87.06 / 60.07 (T / T)	91.58 / 77.13 (W / W)	80.10 / 56.62 (T / T)

masking keeps the positions, it saves no decoder FLOPs at all: the masking variant costs the full 8.477 TFLOPs at every budget. The FLOPs we quote for FastV in Table 18 are therefore for its token-dropping variant, which selects the same tokens but removes them. At $K = 8$ that variant is arithmetically identical to SHAPRUNE after layer 8 — same token-layers, same 3.350 TFLOPs — so the decoder cost is matched; what remains different is the selection rule itself (single tokens against block averages), FastV's re-selection at every decoding step, the masking, and FastV's own environment. We also kept FastV's shipped `fast_v_sys_length` of 36, although in our prompt the image starts at position 35, so FastV never drops the first image token and ranks the first text token after the image among its candidates, masking it after layer K when it is not kept; its effect is measured by the corrected re-implementation of Table 21. It leaves the FLOPs unchanged ($62 + k$ positions either way), but whenever that text token is not kept the token-dropping variant carries 3,102 rather than 3,072 visual token-layers at $K = 2$ and 6,168 rather than 6,144 at $K = 8$. The effect is not confined to one of the k kept positions: the first image token is kept in addition to them, and one prompt token can be masked in every layer from K on. The two TextVQA cells FastV wins at layer 8 (§6) are wins of this masking variant, which we ran without a key–value cache, as

FastV's A-OKVQA evaluation script does, and which re-selects at every decoding step (the in-place token-dropping branch of FastV's LLaVA code supports no cache; its separately shipped variants prune a cached prompt once); a pruner that caches keys and values, as ours does, selects once. Our own pruner keeps the original position ids after layer 8, so its outputs stay consistent with what the masking variant would produce.

FastV's score applied once. To see how much of FastV's TextVQA advantage depends on its uncached path, we applied its score once, in our harness: the last prompt token's attention in layer 7, averaged over heads, over the 576 image positions starting at the true image start, the top k single tokens kept, the layer-0–7 keys and values sliced to them and the answer decoded from the cache, exactly as SHAPRUNE is run. On 1,000 POPE questions, whose one-token answers leave nothing to re-select, it returns the official run's answer on 99.7–99.9% of questions at every budget. On TextVQA it keeps 88.2, 83.7 and 75.0% of answers at 192, 128 and 64 tokens: SHAPRUNE ties it at 192 and 128 ($p = 0.30$ and 0.36) and keeps more at 64 ($p = 10^{-10}$), and the official run keeps 1.6, 2.1 and 3.2 points more than it (1.5, 2.1 and 3.1 against a common reference; each a win for the official run). That difference cannot be assigned to re-selection: besides re-selecting, the official run also lets every decoding step attend in layers 0–7 over all 576 image tokens, recomputes the prompt in layers 8 and above, keeps one image token more (the sys-length offset above) and answers against its own environment's reference.

FastV with the corrected image boundary on POPE, GQA and MME. The same once-applied re-implementation, with the image starting at its true position, was run on the full POPE, GQA and MME sets (Table 21). No difference is detected between it and the official run at layer 8 in any of the 18 recall and accuracy cells (e.g. POPE recall at 64 tokens 88.10 against 88.18% on the distinct pairs, $p = 0.34$), so the sys-length offset does not change FastV's answers measurably. Against it, SHAPRUNE after layer 8 keeps more answers or is more accurate in every cell except four ties: POPE accuracy at 192 tokens, MME recall at 192 and MME accuracy at 192 and 128. As a diagnostic re-implementation it is not counted in the equal-token tally of §5.3, and, as above, no difference between the two FastV paths is attributed to re-selection.

Table 21: SHAPRUNE after layer 8 against FastV's score with the corrected image boundary, applied once through our cached path at layer 8. Full sets; POPE rates and tests on the 5,170 distinct pairs. w–ℓ: questions only SHAPRUNE / only FastV gets right (recall: keeps the un-pruned answer); the 99% image-level interval is SHAPRUNE minus FastV, in points.

Bench.	k	Metric	SHAPRUNE	FastV once	w–ℓ	p	99% interval
POPE	192	recall	98.05	95.96	167–59	3.9×10^{-13}	$[+1.33, +2.85]$
POPE	192	acc.	89.54	89.61	111–115	0.84	$[-0.81, +0.66]$
POPE	128	recall	97.45	93.15	307–85	1.4×10^{-30}	$[+3.36, +5.25]$
POPE	128	acc.	89.56	88.28	229–163	0.001	$[+0.33, +2.27]$
POPE	64	recall	96.32	88.10	538–113	3.4×10^{-67}	$[+6.98, +9.47]$
POPE	64	acc.	89.25	84.24	455–196	1.2×10^{-24}	$[+3.83, +6.18]$
GQA	192	recall	92.31	86.29	1210–452	8.5×10^{-80}	$[+4.98, +7.12]$
GQA	192	acc.	61.31	59.16	725–455	3.7×10^{-15}	$[+1.31, +3.00]$
GQA	128	recall	89.71	80.00	1744–523	1.2×10^{-152}	$[+8.55, +10.92]$
GQA	128	acc.	61.09	57.39	1060–595	1.6×10^{-30}	$[+2.71, +4.70]$
GQA	64	recall	84.58	69.92	2498–654	6.3×10^{-252}	$[+13.18, +16.13]$
GQA	64	acc.	60.14	52.50	1688–727	2.6×10^{-87}	$[+6.38, +8.96]$
MME	192	recall	96.00	95.16	76–56	0.098	$[-0.42, +2.15]$
MME	192	acc.	78.94	79.28	62–70	0.54	$[-1.56, +0.88]$
MME	128	recall	94.73	92.80	114–68	8.0×10^{-4}	$[+0.46, +3.50]$
MME	128	acc.	79.15	78.05	104–78	0.064	$[-0.34, +2.53]$
MME	64	recall	91.87	85.68	251–104	3.7×10^{-15}	$[+4.17, +8.26]$
MME	64	acc.	78.48	72.87	244–111	1.3×10^{-12}	$[+3.66, +7.46]$

Memory. On the 611-position stand-in used for the timings, fp16 and batch 1, the memory allocated above the 13.8 GiB resident before the forward pass (fp16 weights, with LLaVA's loader leaving

the CLIP vision tower in fp32) peaks at 368.1 MiB un-pruned, at 64.5, 109.3 and 140.0 MiB when pruning to 64, 128 and 192 tokens before the LLM, and at 185.4 MiB after layer 8 at every budget, a peak reached in layers 0–7 before any token is dropped. The key–value cache that remains for decoding is the same for the two placements, 55.0, 87.0 and 119.0 MiB, because after layer 8 the cache of layers 0–7 is sliced to the kept positions; un-pruned it is 305.5 MiB. These are allocated, not reserved, figures. Because the decoding cache is the same, each decoded token after the first costs the same under the two placements, and the difference is the one-time prefill, whose share shrinks as answers grow longer; we did not measure long generations.

H Hyperparameters

Labels. 1,400 VQAv2 train2014 image–question pairs. For each pair, 512 coalitions over the 64 blocks are drawn from the Shapley kernel, each paired with its complement, giving 1,024 coalition evaluations and so, with the full and empty coalitions that the efficiency constraint needs, 1,026 forward passes, seeded by a hash of the pair id; $v(S)$ is the log-probability of the first informative token of the greedy un-pruned answer (at most 16 tokens; conditioned on a leading word-boundary token where the answer begins with one) with the blocks outside S removed. Block values are the efficiency-constrained KernelSHAP solution with a ridge of 10^{-6}. Figure 6 follows one training pair through the label computation and the regression. Computing the labels took 58.76 GPU-hours on one RTX 3090 (the single-ablation set 3.73), and training the head about 136 s on a CPU. As SHAPrune is not faster end to end than an optimized DPP at its depth or than pruning before the LLM with pure DPP or an optimized selection (§5.5), we find no request count at which this offline cost is recovered against those (a tentative estimate).

Attention head. $\theta \in \mathbb{R}^{256}$ initialized to $\mathbf{1}/256$ and $b = 0$ (the stored b is -0.13; Pearson correlation is invariant to an additive shift, so b is not counted and we read nothing into it); AdamW with learning rate 10^{-2}, weight decay 0.01 and a one-cycle schedule; batches of 32 pairs; 30 epochs; gradient clipping at 1.0. The loss is one minus the Pearson correlation between the 64 block-averaged scores and the 64 block labels, averaged over the pairs of a batch. Performance saturates early in the number of training pairs:

Training pairs (120 more held out)	175	350	700	1,280
Val. Pearson (120 pairs)	0.275	0.272	0.276	0.276
POPE dev. recall / acc. at 64	94.7 / 87.6	94.5 / 87.2	94.2 / 87.1	94.1 / 87.0

I Diagnostics

A semivalue that re-weights the coalitions (registered). The difference-of-means and least-squares controls of Table 13 estimate positive affine transforms of the Shapley value and so do not test its weighting. We therefore re-solved the same 1,024 coalition evaluations per pair as the Beta(16,1) semivalue of Kwon & Zou (2022), which weights small coalitions heavily (a solver checked against exact enumeration on 8-player games), trained the identical head with the same recipe and seed, and evaluated it after layer 8 at 64 tokens with recall on POPE (de-duplicated) and GQA as the registered decision cells. The Beta labels are more reproducible than the Shapley labels (split-half $+0.749$ against $+0.519$). No difference from SHAPrune is detected: POPE 96.36 against 96.32% of the 5,170 distinct pairs (95.77 against 95.87% of the 9,000 rows; 27–25, $p = 0.89$, 99% interval $[-0.31, +0.42]$), GQA 84.43 against 84.58% (243–262, $p = 0.42$, $[-0.66, +0.37]$). A tie is not equivalence; the 192- and 128-token cells and accuracy were not run.

Protocol. All diagnostics use the 120 held-out training pairs, whose block Shapley labels serve as the reference. *Label agreement* is, averaged over pairs, the Spearman correlation between a selector's 64 block scores and the labels, and the overlap of their top-7 blocks (chance $7/64 = 0.109$). The *question swap* pairs each of 60 held-out images with the question of another held-out pair under a fixed random derangement, recomputes the selector's input for the swapped prompt — for SHAPrune this re-runs the first 8 layers, so the attention features themselves change — and reports the fraction of the original top-7 blocks that survive; 1.0 means the question is ignored. On these pairs SHAPrune's top-7 blocks agree with the labels' at 0.363 (chance 0.109; Spearman correlation of the 64 block scores $+0.139$), and 48.6% of them survive a question swap: about half of the selected blocks change with the question. This shows that the selection depends on the question, not that the new selection suits the new question, which we did not evaluate; and the shuffled-label control of §5.4 permutes image and question together, so it does not separate their contributions (a shuffle of questions within an image would; not run). Against single-ablation labels the trained head's block scores reach Spearman $+0.04$ (input removal) and $+0.08$ (deployed edit) on these pairs; its training target is the 1,024-coalition Shapley labels, against which it reaches $+0.14$, and those labels themselves agree with single-ablation labels only at $+0.19$ and $+0.22$.

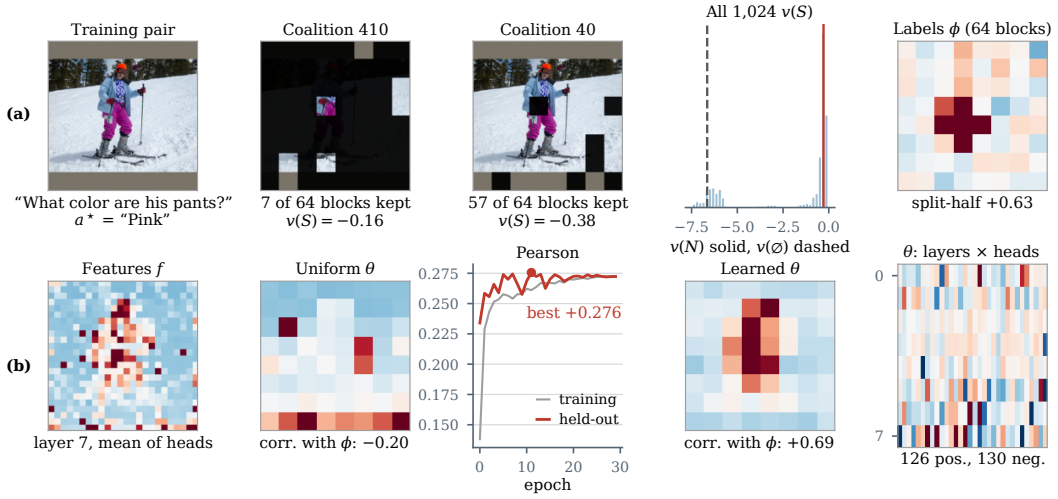

Figure 6: Training on one pair, drawn from the stored cache, labels, checkpoint and training log; the pair was chosen for legibility. **(a)** Labels, computed once per pair with 1,026 forward passes of the frozen model: a VQAv2 training pair and the model's own answer; the two coalitions at the 25th and 75th percentiles of coalition size: keeping 7 of 64 blocks scores higher ($v(S) = -0.16$) than keeping 57 (-0.38) although on this pair 7-block coalitions average -3.03 and 57-block ones -0.30 (coalition 410 keeps the block with the second-largest ϕ and drops the one with the largest; coalition 40 does the opposite); all 1,024 values, from -7.46 to -0.04, with $v(N) = -0.29$ (solid) and $v(\emptyset) = -6.65$ (dashed) marked, 229 of them above the full image; and the block Shapley values that KernelSHAP solves from them (split-half reliability $+0.63$ on this pair). **(b)** Regression, which only reads those results: the attention features (for display, layer 7 averaged over its 32 heads); the block scores of uniform θ, highest in the bottom block row, which on this image is LLaVA's square padding (correlation with ϕ: -0.20); the Pearson correlation over 30 epochs, a running value on the 1,280 training pairs and the value on the 120 held-out pairs, with the checkpoint kept at epoch 11, counting from 0 ($+0.276$ on the held-out pairs); the block scores of the learned θ ($+0.69$, in-sample, at the 94th percentile of the 1,280 training pairs, whose median is $+0.31$ against -0.02 for uniform θ); and the 256 head weights, 126 positive and 130 negative. Score grids have their mean subtracted and ϕ is centered at zero; each grid has its own color scale, clipped at the 96th percentile of its absolute values. Neither changes a ranking or a top-k selection.

Input removal against the deployed edit. The labels of §3.5 are the single-ablation block labels of the 120 held-out pairs recomputed under the deployed edit (layers 0–7 over all 576 tokens, the block's keys and values dropped from layer 8 on, original position ids). Per pair they correlate with the input-removal labels at Pearson $+0.73$ on average (median $+0.78$; Spearman $+0.42$) and share 50% of their top-7 blocks (chance 11%). The reference, two computations of the deployed labels on 30 of the pairs that differ only in the batch shape of the fp16 forward passes, agrees at Pearson $+0.98$ (Spearman $+0.84$, top-7 81%); repeating one computation at the same batch shape reproduces it exactly. The Pearson ratio, 0.75, is the registered criterion; the Spearman ratio, 0.50, was computed afterwards, so the two labelings agree less in rank than in value. The deployed-edit effects are about half as large (median per-pair mean $|\delta_i|$ 0.006 against 0.012 nats). *Retraining on them* (registered before the run): we computed the deployed-edit labels for a fixed subset of 350 of the 1,280 training pairs (validation on the 120 held-out pairs above) and trained the head on them with the recipe and seed of SHAPRUNE, and a second head on the same pairs' input-removal single-ablation labels. After layer 8 at 64 tokens no recall difference is detected between the two on POPE (96.69 against 96.52% of the 5,170 distinct pairs, 38 against 29 discordant pairs, $p = 0.33$, 99% interval $[-0.21, +0.56]$) or GQA (83.19 against 83.71%, 352 against 417, $p = 0.021$, $[-1.12, +0.09]$). SHAPRUNE, trained on 1,400 pairs of Shapley labels, ties the deployed-edit head on POPE ($p = 0.076$) and keeps more GQA answers (84.58 against 83.19%, $p = 3 \times 10^{-8}$).

The mean-map head. This head has SHAPRUNE's features, pairs, recipe and seed, but every pair's label is replaced by one fixed vector, the element-wise mean of all 1,400 labels, so its supervision

carries no pair-specific content; it still reads each question's own attention. After layer 8 at 64 tokens it keeps 86.0% of POPE answers (89.98% of the distinct pairs) and 75.4% of GQA answers. SHAPRUNE keeps more on both (443 against 115 discordant pairs, $p = 2 \times 10^{-46}$, $[+5.2, +7.6]$; 1,842 against 688, $p = 1 \times 10^{-120}$, $[+7.9, +10.5]$), and so does the shuffled-label head on POPE (91.55 against 89.98%, 154 against 73, $p = 8 \times 10^{-8}$, $[-2.36, -0.79]$ for the mean-map head); on GQA the two tie (75.41 against 76.37%, $p = 0.0014$, $[-1.97, +0.01]$). The comparisons were registered before the runs.

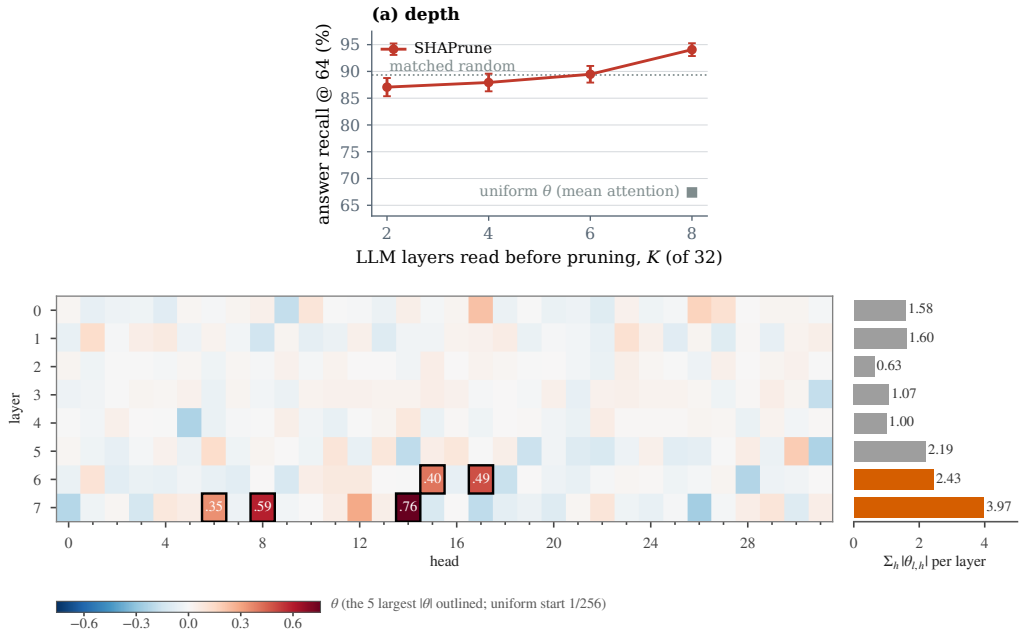

Figure 7: The Shapley-supervised head. **(a, top)** Answer recall at 64 tokens against the number of LLM layers K whose attention the head reads (development subset, pruning before the LLM; bars are 95% binomial intervals; the axis starts at 63%). The gray square is the same head with θ left uniform, which reduces it to the average attention over all heads. **(b, bottom)** The 256 learned head weights θ as 8 layers $\times$ 32 heads, on a diverging scale centered at zero; the five largest, all positive and in layers 6 and 7, are outlined with their values (uniform start $1/256 \approx 0.004$). Right: $\sum_h |\theta_{l,h}|$ per layer, 44% of the total in layers 6–7. The weights are signed (126 positive, 130 negative; $\sum \theta = 0.47$ against $\sum |\theta| = 14.47$). The Pearson objective is invariant to the scale of θ, so only ratios between weights carry meaning.

Where the learned signal lives. Heads restricted to the first 2, 4 or 6 layers tie matched random on the development subset (87.1, 87.9 and 89.5% against 89.3%), while 8 layers give 94.1% (Figure 7a); layer 6 alone gives 93.5% and layer 7 alone 93.9%, and the learned weight mass concentrates there ($\sum |\theta|$ of 3.97 at layer 7 against 0.63 at layer 2; Figure 7b). Early-layer text-to-image attention is sink-dominated (Kang et al., 2025); the Shapley labels select deeper heads.

A per-layer head (registered). To test whether SHAPRUNE's 256 head-level weights matter beyond which layers they read, we trained, on the same labels and with the same recipe, a head with one weight per layer shared by its 32 heads (8 weights and a bias; best epoch 5, held-out Pearson $+0.236$ against $+0.276$ for SHAPRUNE), and registered before its runs that *answer recall* at 64 tokens after layer 8 on POPE and GQA decides. SHAPRUNE keeps more answers on both: 96.32 against 95.42% on the 5,170 distinct POPE pairs (95.87 against 94.56% on the 9,000 rows; 124–77, $p = 1.1 \times 10^{-3}$, $[+0.17, +1.65]$) and 84.58 against 81.84% on GQA (891–546, $p = 7.6 \times 10^{-20}$, $[+1.80, +3.71]$). The per-layer weights (descriptive) are within ± 0.021 of zero for layers 0–3 and 5, -0.198 for layer 4, and $+0.101$ and $+0.139$ for layers 6 and 7.



Figure 8: Inference on three of the 120 held-out pairs, chosen as the three whose block-level Pearson correlation with ϕ rises most from uniform to learned θ (their answers are distinct; the middle row, the example of Figure 1, rises from Pearson -0.03 to $+0.79$); over all 120 held-out pairs the median Pearson correlation is -0.01 for uniform θ and $+0.30$ for the learned head (mean $+0.28$, the validation value of §3.3). Columns: the image on the 24×24 token grid; the scores of uniform θ, every head at $1/256$ (the untrained control of Figure 2), whose top 64 after block averaging (orange outlines, second column) lie mostly in the bottom block row (6, 4 and 4 of their 7 whole blocks), where the scores are 2.0–3.4 times those of the top row; the learned score $s = \theta^\top f$, whose top 64 after block averaging (green outlines, third column) share 3, 5 and 6 of their 7 whole blocks with the 7 blocks of largest ϕ, against none for uniform θ; and those 64 tokens, which SHAPRUNE keeps after layer 8. The numbers under the learned score split it into the contributions of the positive and the negative head weights ($\times1000$) for two tokens, the most attended token of the bottom row and the highest-scoring token in the block of largest ϕ (top-ϕ), with the share of the positive part that the negative part cancels in parentheses. Because the weights nearly cancel ($\sum\theta = 0.47$ against $\sum|\theta| = 14.47$), a signal spread evenly over the heads loses 94% of its positive part; the bottom-row token loses about that much (90–99%) and the top-ϕ token less (24–51%). Part of that gap comes from picking the highest-scoring token of a block, which by itself lowers the share (a median of 78% over every block's highest-scoring token on the 120 held-out pairs); on block-mean features, the unit the pruner ranks, the medians are 72% for the top-ϕ block, 90% for the median block and 102% for the bottom-row blocks. In all three rows the most attended bottom-row token lies in the square padding LLaVA adds to non-square images. Overlays are token scores with their mean subtracted, clipped at the 95th percentile. No Shapley value is computed at inference; θ was fixed in training.

How similar the trained heads are (registered, descriptive). Registered before computing, with numbers only as the outcome: on the 120 held-out pairs we rank the 64 block scores of each head (no padding exclusion or tie-break) and compare them with SHAPRUNE's. Two training seeds agree at a median per-pair Spearman of $+0.967$ (IQR $[0.953, 0.979]$) with a mean top-7-block overlap of 0.948. Heads trained on other labels agree at $+0.982$/0.936 (Banzhaf-estimator labels), $+0.957$/0.918 (difference of means), $+0.913$/0.840 (175 pairs), $+0.677$/0.805 (single ablation), $+0.618$/0.719 and $+0.601$/0.757 (the two 350-pair heads of Appendix I), $+0.723$/0.463 (shuffled labels) and $+0.703$/0.407 (mean-map labels); the HAWK-style weights agree at $+0.057$/0.020 and uniform θ at $+0.029$/0.039. A per-layer weighting of the same features (SHAPRUNE's θ averaged within each layer, 8 parameters) agrees at $+0.404$/0.349, and $\sum \theta / \sum |\theta| = +0.032$ for SHAPRUNE (-0.06 to $+0.04$ for the other trained heads). The weight vectors themselves agree with SHAPRUNE's at cosine $+0.995$ (second seed), $+0.976$ (Banzhaf), $+0.946$ (difference of means) and $+0.702$ (single ablation).

Where the weight goes. Summed over the 32 heads of each layer, $\sum |\theta|$ is 1.58, 1.60, 0.63, 1.07, 1.00, 2.19, 2.43 and 3.97 for layers 0 to 7 (Figure 7b, right). The five heads with the largest weights are (layer, head) $= (7, 14)$, $(7, 8)$, $(6, 17)$, $(6, 15)$ and $(7, 6)$, with $\theta = 0.758$, 0.587, 0.488, 0.404 and 0.349 against the uniform initial $1/256 \approx 0.004$ (the Pearson objective is invariant to a positive rescaling of θ, so only the ratios between weights carry meaning, not their scale). The weights are signed — 126 positive and 130 negative, summing to 0.47 against $\sum |\theta| = 14.47$ — so attention spread evenly over the heads largely cancels. Measured on blocks, the unit the pruner ranks, the block the labels rank first keeps more of its positive part than the others: on the 120 held-out pairs a median of 72% of it is canceled, against 90% for the median block, while the bottom-row blocks lose more than their positive part (102%). Figure 8 shows single tokens on three pairs chosen for contrast.

Fit and recall (post hoc). Over 18 heads evaluated on the development subset at 64 tokens, recall has a Spearman correlation of $+0.538$ ($p = 0.021$) with the validation Pearson correlation of the fit and $+0.598$ ($p = 0.009$) with its top-7 overlap. The relation comes mostly from the depth axis ($+0.879$ over the nine heads that differ in the layers they read); over the ten heads that differ in their labels it is -0.305, not significant. This analysis was done after the fact on a small sample and is tentative. That the head does more than correct the positional bias rests on the registered controls after layer 8 (POPE de-duplicated / GQA recall at 64 tokens): uniform θ over layers 0–7 keeps 78.4/58.8%, over layers 6–7 88.7/69.5%, the shuffled-label head 91.5/76.4% and SHAPRUNE 96.3/84.6%, each step significant ($p < 2.5 \times 10^{-10}$); the mean-map head, whose supervision is a pure spatial prior, keeps 90.0/75.4% (paragraph above).

J Additional benchmarks: MME and TextVQA

MME and TextVQA are run with the protocol of Appendix A, using the same pruning methods and code as the main table with only the question files changed. The harness reproduces the published numbers to within 1%, although VisionZip's MME score falls below the published one at every budget. Un-pruned: MME perception 1504.6 and P+C 1862.4 (published 1510.7 and 1862), TextVQA 58.23 (58.2). VisionZip at 192 / 128 / 64: MME P+C 1769.0 / 1757.5 / 1687.6 (published 1782.6 / 1761.7 / 1690), TextVQA 57.24 / 56.83 / 55.55 (57.3 / 56.8 / 55.5). CDPruner at 128 / 64: MME perception 1437.2 / 1403.4 (1431.4 / 1415.1), TextVQA 56.19 / 55.14 (56.2 / 55.3). FastV publishes no pruned number at our budgets (its paper's grid, $K \in \{0, 2, 3, 5\}$ by $R \in \{50, 75, 90\}$%, keeps about 288, 144 or 58 tokens, and its README's LLaVA-1.5 table uses $K = 3$, $R = 50$%), so there is no pruned FastV number to reproduce; its environment reproduces the un-pruned A-OKVQA accuracy in its README, 76.9 against 76.8 (76.7 in its paper). FastV's released code fixes its rotary table at 1,000 positions, which 5 of the 5,000 TextVQA prompts exceed; we extend it to the sequence length at run time, so every sequence of at most 1,000 positions, which includes all of POPE, GQA and MME, is processed bit-identically.

Paired tests. Before the LLM the picture of Appendix F repeats. SHAPRUNE before the LLM beats matched random on recall at every budget on both benchmarks ($p \leq 5 \times 10^{-5}$ on MME, $p < 10^{-25}$ on TextVQA). Against CDPruner it ties everywhere except a win on MME accuracy at 64 tokens ($p = 0.006$). Against pure DPP it ties on MME and loses on TextVQA at 64 tokens (recall: 371 against 494 discordant questions, $p = 3 \times 10^{-5}$; accuracy $p = 0.0065$), and it trails

Table 22: MME (2,374 questions): recall / P+C score. TextVQA (5,000): recall / accuracy. The rows down to SHAPRUNE before the LLM prune before the LLM; FastV prunes inside the model; it is run from its official code and scored against its own environment's un-pruned answers (Appendix A); pure DPP moved to layer 8 is run by us with SHAPRUNE's reuse of layers 0–7, its similarities still computed on the projector outputs as in CDPruner, so only the pruning point moves; the last row is SHAPRUNE after layer 8, as in Table 12. Bold marks the best pruned recall or accuracy in each column.

Method	192	128	64
MME — recall / perception + cognition			
Un-pruned (576)	100 / 1862.4	100 / 1862.4	100 / 1862.4
Matched random	89.6 / 1730.5	87.0 / 1661.7	82.4 / 1535.8
VisionZip	92.5 / 1769.0	91.4 / 1757.5	87.5 / 1687.6
CDPruner	92.2 / 1783.8	89.9 / 1775.4	87.1 / 1709.8
Pure DPP	92.3 / 1771.2	90.0 / 1738.0	87.8 / 1683.9
DivPrune	91.6 / 1765.8	89.6 / 1710.6	86.6 / 1615.4
SHAPRUNE before the LLM	91.9 / 1789.4	90.2 / 1777.8	87.2 / 1763.3
FastV, after layer 2	94.9 / 1840.9	92.2 / 1774.0	85.5 / 1557.5
FastV, moved to layer 8	95.2 / 1850.5	92.8 / 1781.0	85.4 / 1527.7
Pure DPP, moved to layer 8	95.9 / 1830.4	94.0 / 1792.4	**92.2** / 1733.6
SHAPRUNE after layer 8	**96.0** / 1832.1	**94.7** / 1851.3	91.9 / 1850.7
TextVQA — recall / accuracy			
Un-pruned (576)	100 / 58.23	100 / 58.23	100 / 58.23
Matched random	71.9 / 52.83	67.1 / 50.83	62.0 / 48.27
VisionZip	82.0 / 57.24	79.6 / 56.83	73.6 / 55.55
CDPruner	81.2 / 57.31	77.9 / 56.19	73.1 / 55.14
Pure DPP	81.5 / 57.36	78.4 / 56.52	74.3 / 55.63
DivPrune	80.1 / 56.62	77.7 / 55.95	73.3 / 54.55
SHAPRUNE before the LLM	81.4 / 57.75	77.7 / 56.73	71.8 / 54.50
FastV, after layer 2	87.9 / 58.01	84.0 / 57.04	76.4 / 55.01
FastV, moved to layer 8	**89.8** / 57.98	**85.8** / 57.64	78.2 / 55.07
Pure DPP, moved to layer 8	87.5 / 57.66	84.8 / 57.35	**79.6 / 56.37**
SHAPRUNE after layer 8	87.7 / **58.28**	84.2 / **57.85**	79.0 / 56.08

VisionZip on TextVQA recall at 128 and 64 ($p \leq 0.003$). After layer 8, SHAPRUNE beats all five pre-LLM baselines, DivPrune included, on recall at every budget on both benchmarks ($p < 10^{-7}$), and its MME score at 64 tokens, 1850.7, is within 12 points of the un-pruned model. Among the published methods, only FastV keeps as many answers as SHAPRUNE at equal kept tokens on these benchmarks: it ties SHAPRUNE on MME at 192 tokens at both layers and on TextVQA at 192 and 128 tokens at layer 2, and at layer 8 it keeps more TextVQA answers at 192 and 128 tokens (§6); at 64 tokens SHAPRUNE keeps more answers than FastV at either layer on both benchmarks, except against FastV at layer 8 on TextVQA, a tie. Pure DPP moved to layer 8 ties SHAPRUNE on MME recall and per-question accuracy at every budget, although its score is lower at 128 and 64 tokens (1792.4 and 1733.6 against 1851.3 and 1850.7); on TextVQA recall it ties it at every budget. Moving pure DPP from before the LLM to layer 8 adds 3.5, 4.0 and 4.4 MME and 6.0, 6.4 and 5.3 TextVQA recall points at 192, 128 and 64 tokens. The MME score differences between methods are mostly not borne out by the paired tests on per-question accuracy, and we do not rank methods by the score.

The official MME score against the per-question tests (post hoc). The official score adds, for each of MME's 14 subtasks, per-question accuracy and acc+ (the share of images with both of their questions right), each out of 100, so every subtask weighs 200 points whatever its size: a 20-image subtask (OCR, code reasoning, numerical calculation, text translation) counts as much as a 200-image one. Recomputed with a re-implementation of the official evaluator that reproduces its totals exactly, SHAPRUNE after layer 8 scores 117.1 points above pure DPP at layer 8 at 64 tokens (1850.7 against 1733.6), 99% image-bootstrap interval $[+14.8, +218.3]$ (10,000 resamples, seed 0); at 128 and 192 tokens the differences, $+58.9$ and $+1.7$, have 99% intervals containing zero. At 64 tokens the four 20-image subtasks alone differ by $+32.5$ (OCR), $+27.5$ (code reasoning), $+25.0$ (numerical calculation) and $+7.5$ points (text translation), while the per-question tests detect no difference at

any budget (at 64 tokens recall 91.87 against 92.17%, $p = 0.67$, and accuracy 78.48 against 77.76%, $p = 0.26$), nor does a paired image-level test of acc+ pooled over all 1,187 images (60.57 against 59.56% at 64 tokens, $p = 0.36$). The score thus weights a few small subtasks heavily and is not what our per-question tests test. SHAPRUNE's non-monotone score across budgets (1832.1, 1851.3 and 1850.7 at 192, 128 and 64 tokens) is within bootstrap noise: the 99% interval of every pairwise difference contains zero (192 minus 64 tokens: $[-103.3, +65.5]$).

K Progressive in-LLM pruners: SparseVLM and PyramidDrop

SparseVLM (Zhang et al., 2025c) and PyramidDrop (Xing et al., 2025) prune in stages at several layers of the language model rather than once. Both run from their official code in their own environments, with our question files, prompt, greedy decoding and scoring, and each is scored against the un-pruned answers its own environment produces, as FastV is (Appendix A).

Reproduction. SparseVLM's LLaVA-1.5-7B configuration retains 192, 128 or 64 tokens. Our run gives POPE category-mean F1 85.44 / 85.03 / 77.58 against the published 85.3 / 85.0 / 77.5 and GQA accuracy 59.5 / 58.4 / 53.7 against 59.5 / 58.4 / 53.8, inside the 1.0-point margin we fixed before running it, and MME score 1778.1 / 1763.3 / 1600.6 against 1787 / 1746 / 1589; our POPE file has 9,000 questions where the authors' has 8,910. PyramidDrop's repository ships a single schedule for the 7B model, averaging 270 tokens per layer, and its authors state in the issue tracker that the settings for other budgets were not recorded, so there is no official 192- or 128-token configuration; we did not construct one. The shipped schedule reproduces the published numbers: POPE category-mean F1 84.80 and GQA accuracy 60.14 against the published 84.8 and 60.1. We ran the schedule its authors recommend for an average of 64 tokens per layer (drops at layers 2, 8 and 16 with $\lambda = 0.19$, a layer average of 62.94 visual tokens), on all four benchmarks. It does not reproduce the numbers PyramidDrop publishes for an average of 64, whose schedule was not recorded: its GQA accuracy is 44.97 against 47.5, its TextVQA accuracy 47.82 against 50.6 and its MME score 1255.5 against 1561, so this row is the authors' recommended substitute rather than a reproduction.

What is and is not matched. These methods state their budget as an average over layers, and Table 23 compares at that nominal budget. That does not equalize compute, and the inequality favors us: at a nominal 64 SparseVLM costs 2,632 visual token-layers and PyramidDrop 2,014, against our 6,144 after layer 8 (Table 18), so we spend more than twice their visual token-layers, and 1.8 and 2.0 times their decoder FLOPs (1.9 for PyramidDrop with its ranking pass counted), for these wins. The comparison that equalizes compute is Table 19, where SparseVLM at 192 tokens — which spends 5% more decoder FLOPs than our 64 after layer 8 — ties us on POPE and MME and beats us on GQA and TextVQA.

Verdicts at the same nominal budget. SHAPrune after layer 8 keeps significantly more answers than SparseVLM in all twelve of its benchmark–budget cells and than PyramidDrop in all four of its, by the rule of §4; we count these sixteen cells separately from the same-kept-tokens comparisons, because their budget is a layer average and ours is not; over the other methods and budgets of this paper those comparisons come to 114 recall cells, 83 won, 26 tied (FastV at layers 2 and 8 on MME at 192 tokens, at layer 2 on TextVQA at 192 and 128 tokens and at layer 8 on TextVQA at 64, pure diversity moved after layer 8, at every budget on all four benchmarks, and CDPruner moved after layer 8, at every budget on POPE, MME and TextVQA) and 5 lost (FastV moved to layer 8, on TextVQA at 192 and 128 tokens, and CDPruner moved to layer 8, on GQA at all three budgets). On accuracy SHAPrune after layer 8 is more accurate than SparseVLM in eight of the twelve cells and ties the other four (POPE at 192 and 128 tokens, MME and TextVQA at 192), and more accurate than PyramidDrop in all four ($p \leq 3 \times 10^{-48}$); no cell is lost. PyramidDrop's MME score at 64 is 1255.5 (perception 948.8, cognition 306.8), against 1862.4 for its own un-pruned run.

Matching the layer average instead. Because these methods count their budget as an average over layers, before running them we fixed one cross-budget pairing that matches that average: our 64 tokens after layer 8, which is a layer average of 192 visual tokens, against SparseVLM at 192, which is 192.03 by its own counter and 206.50 if one counts the tokens each layer actually processes. There we tie on POPE recall (95.9 against 95.2, $p = 0.20$), on POPE accuracy and on GQA accuracy, and *lose* on GQA recall: 84.6 against 85.9, $p = 10^{-4}$, 99% interval $[-2.40, -0.29]$. The same pairing on the two benchmarks added later ties on MME and loses on TextVQA (Table 19). We report this as it was registered.

Table 23: Progressive in-LLM pruners at their nominal (layer-average) budget. Recall / accuracy on the full sets; accuracy is the per-question rate, so the MME column is *not* the official perception + cognition score used in Tables 3 and 22 and the two are never read off each other (Appendix A). TextVQA accuracy is rounded to one decimal here. PyramidDrop, which has no official 192- or 128-token setting and was run only at its recommended schedule averaging 64 tokens, is in Table 3. Every recall cell of SHAPRUNE after layer 8 in this table is a win by the rule of §4.

Method	192	128	64
POPE 9,000 — recall / accuracy			
SparseVLM (official)	95.2 / 86.6	95.0 / 86.4	88.0 / 81.0
SHAPRUNE (after layer 8)	97.9 / 86.8	97.3 / 87.0	95.9 / 86.6
GQA 12,578 — recall / accuracy			
SparseVLM (official)	85.9 / 59.5	82.3 / 58.4	72.1 / 53.7
SHAPRUNE (after layer 8)	92.3 / 61.3	89.7 / 61.1	84.6 / 60.1
MME 2,374 — recall / per-question accuracy			
SparseVLM (official)	93.3 / 77.8	92.0 / 77.4	84.5 / 73.3
SHAPRUNE (after layer 8)	96.0 / 78.9	94.7 / 79.1	91.9 / 78.5
TextVQA 5,000 — recall / accuracy			
SparseVLM (official)	84.3 / 57.7	79.9 / 56.7	69.2 / 53.5
SHAPRUNE (after layer 8)	87.7 / 58.3	84.2 / 57.9	79.0 / 56.1

L A SECOND BACKBONE: QWEN2.5-VL-7B

Limits of the replication. We do not address how to define the 3×3 blocks on grids whose shape varies with the aspect ratio, and neither the square padding nor the label noise is isolated as a cause of the weaker Qwen results.

We repeated the whole pipeline — block Shapley labels, the linear head, the evaluation — on Qwen2.5-VL-7B (Bai et al., 2025), with four departures from that model's own practice or from our LLaVA protocol: the image grid, the placement, the position ids and the decoding. The image is square-padded to 672×672 pixels and encoded into 576 visual tokens on a 24×24 grid, so that the 3×3 block game, the budgets and the matched random subsets are the ones used for LLaVA. Qwen's own dynamic-resolution preprocessing gives 276–529 tokens on the images we measured, so this is not the setting the Qwen literature evaluates and the numbers here are not comparable with published Qwen results. Pruning in Table 24 is before the LLM, the placement of Table 15; the last paragraph and Table 25 move it after layer 8. Pruned sequences keep Qwen's original M-RoPE position ids rather than being re-indexed, since re-indexing would scramble its two-dimensional grid; this is the one place where the Qwen game differs from LLaVA's (§3.1). The head reads the first 8 of Qwen's LLM layers, 28 heads each, so it has 224 head weights. We generate with pure greedy decoding (`repetition_penalty` 1.0) rather than the checkpoint default of 1.05, for both the labels and the un-pruned reference answers. Before these labels were computed, a 20-pair measurement of label reliability ($+0.22$ against $+0.46$ for LLaVA on the same pairs) had fallen below a threshold we had set for continuing; because that threshold had been fixed after seeing part of the data and had never been related to downstream performance, we withdrew it and ran the downstream test below instead, with its success criterion registered before the labels were computed.

The labels are noisier. We computed labels for 820 pairs, skipping none; their split-half reliability is $+0.313$ on average and $+0.278$ at the median. A diagnostic pass over 810 of them gives $+0.312$ and $+0.276$, negative for 15% of pairs, against $+0.515$ and $+0.545$ for LLaVA labels on the same pairs (Wilcoxon $p = 1.2 \times 10^{-52}$). The rest of the diagnostic is clean but not spotless: no missing, non-finite, truncated or empty answers, 78.4% agreement with the VQAv2 ground truth, and a median $v(N) - v(\emptyset)$ of 2.36 nats; but in 5 of 810 pairs the first answer token is no longer the argmax when the full image is scored inside a padded batch, in one pair its log-probability there differs from that of `generate()` by more than 0.1 nats, and label reliability differs between the two workers ($p = 0.007$ by a Mann–Whitney test over pairs, $p = 0.019$ by a permutation test over shards). We checked the obvious cause of the last one and ruled it out: recomputing one worker's shard on the other device

reproduced the coalition values bit for bit, so the difference is in which pairs each worker drew, not in the hardware. Qwen and LLaVA labels of the same pair correlate at Pearson $+0.35$ on average, against -0.01 under a shuffled pairing, so the two models do agree about which blocks matter, with a worse signal-to-noise ratio on Qwen's side. Trained on the other 700 pairs, the head reaches a validation Pearson correlation of $+0.187$ with the Shapley labels on the 120 held-out pairs, where uniform θ reaches $+0.096$; the corresponding LLaVA figures are $+0.276$ and $+0.004$.

Results. Table 24 gives results on the full POPE and GQA sets. Against uniform θ — the same features and selection rule with the supervision removed — the trained head wins every cell by the rule of Appendix A except POPE recall at 192 tokens, where the two tie (96.2 against 96.0; 95 against 89 distinct discordant questions, $p = 0.71$). Against a matched random subset it wins every GQA cell and every POPE accuracy cell, but *ties on POPE recall at all three budgets*: at 64 tokens the rates are 93.0 and 90.9, yet on the 5,170 distinct pairs only 258 questions go to the head and 213 to random ($p = 0.043$, 99% interval $[-0.23, +1.95]$), which our rule reports as no difference. By the criterion we registered for this test — the exact McNemar test on the 9,000 rows — it passed ($p = 4 \times 10^{-8}$); by the stricter rule of Appendix A it is a tie, and that is what we report. We therefore claim for Qwen only that Shapley supervision beats unweighted attention in 11 of 12 cells and beats a matched random subset on GQA; on POPE it is more accurate than random but not measurably more faithful, which by the argument of Appendix B is not evidence of a better selection: the un-pruned Qwen model answers *yes* to 40.2% of POPE questions and is 87.5% accurate; the head moves the yes-rate to 43.3, 43.5 and 42.1% at 192, 128 and 64 tokens — and is *more* accurate than the un-pruned model at 192 and 128 — while matched random moves it to 38.5, 37.0 and 34.1% and uniform θ to 39.5, 37.4 and 26.4%. Given a label reliability of $+0.31$ against LLaVA's $+0.52$, a weaker result is what we would expect, but we ran no experiment that would establish the label noise as its cause. Using Qwen's own dynamic resolution would need blocks defined on a grid whose shape changes with the image — for instance a fixed number of regions pooled from the merged patches; we did not try it.

After layer 8. We then ported the placement of §3.5 to Qwen's M-RoPE decoder: layers 0–7 run once on the full sequence through the model's own forward, the head reads the same 224 features as before the LLM, and layers 8–27 run only on the kept tokens, reusing their keys and values with the original M-RoPE position ids. Four arms ran on the full POPE and GQA sets at 192, 128 and 64 tokens: the trained head after layer 8; pure DPP (CDPruner's DPP with constant relevance, on the merger features) after layer 8; the same DPP token sets before the LLM; and the matched random token sets of Table 24 after layer 8. The un-pruned answers equal those of Table 24 on every question. Before any result we registered recall at 64 tokens on POPE and GQA as the deciding cells, the rule of Appendix A, and the sentence each outcome would put in the paper; Table 25 gives every cell. *Placement:* the same pure-DPP token set ties across the two placements (POPE 95.45 against 95.53%, $p = 0.73$; GQA 79.92 against 79.44%, $p = 0.049$, 99% interval $[-0.21, +1.18]$), as do all twelve of its cells, so on Qwen2.5-VL we did not detect a placement effect. The trained head after layer 8 against the same head before the LLM wins POPE recall at 64 tokens (95.30 against 94.68%, 50 against 18 discordant pairs, $p = 1.3 \times 10^{-4}$) and ties the other eleven cells. *Against matched random:* after layer 8 the head also wins both deciding cells (POPE 95.30 against 93.60% on distinct pairs and 93.86 against 90.62% on all 9,000 rows, 263–175, $p = 3.0 \times 10^{-5}$; GQA 79.00 against 76.80%, 1,590–1,314, $p = 3.3 \times 10^{-7}$) and 8 of the 12 cells, tying POPE recall and GQA accuracy at 192 and 128 tokens. *Against pure DPP at the same depth:* no difference is detected at 64 tokens (POPE 95.30 against 95.45%, $p = 0.70$; GQA 79.00 against 79.92%, $p = 0.027$, interval $[-2.31, +0.50]$, the point estimate favoring DPP), so the statement that the two backbones show no difference from same-depth DPP holds at 64 tokens only. On Qwen alone, pure DPP after layer 8 keeps more POPE answers than the head at 192 and 128 tokens (-0.81 and -0.85 points, $p = 0.0016$ and 0.0054; the 128-token interval $[-1.65, -0.04]$ barely excludes zero); on LLaVA the same comparison ties at every budget with the head nominally ahead (Table 8). The remaining GQA recall cells and all six accuracy cells tie. No published pruner was run on Qwen; its comparisons are uniform θ, matched random and pure DPP.

What transferred across backbones. On both backbones the trained head beats unweighted attention (on Qwen in 11 of 12 cells) and, after layer 8, a matched random subset at 64 tokens, and no difference from the same-depth pure DPP is detected at 64 tokens. Two things did not transfer: the placement effect, detected for pure DPP on LLaVA but not on Qwen, and the relation to pure DPP

at 192 and 128 tokens, where on Qwen pure DPP keeps more POPE answers. The Qwen labels are also less reliable ($+0.31$ against $+0.52$); split-half reliability of the labels is the natural candidate for predicting on which backbone the supervision helps, but we did not test it as a cause.

Table 24: Qwen2.5-VL-7B, 576 visual tokens forced, pruning before the LLM. POPE 9,000 and GQA 12,578, *answer recall* / accuracy at 192 / 128 / 64 kept tokens. Uniform θ is the same head with its weights left at $1/224$; matched random is the construction of Appendix A. Bold marks the trained head where it beats *both* controls under the rule of Appendix A. On POPE recall it never does: it ties matched random at all three budgets and uniform θ at 192 tokens.

Method	192	128	64
POPE 9,000 — recall / accuracy			
Un-pruned (576 tokens)	100 / 87.5	100 / 87.5	100 / 87.5
Matched random	95.7 / 85.9	94.1 / 84.6	90.9 / 81.5
Uniform θ (no training)	96.0 / 86.5	93.7 / 84.5	83.8 / 74.6
Trained head	96.2 / **89.0**	95.3 / **88.7**	93.0 / **86.8**
GQA 12,578 — recall / accuracy			
Un-pruned (576 tokens)	100 / 61.1	100 / 61.1	100 / 61.1
Matched random	86.8 / 59.4	82.7 / 58.1	74.9 / 55.4
Uniform θ (no training)	85.1 / 57.5	78.2 / 54.6	66.9 / 49.1
Trained head	**89.4 / 60.4**	**85.9 / 59.4**	**79.0 / 57.2**

M Fresh-image confirmation

The fresh-image set of §5.1 has 5,000 VQAv2 questions, one per image, on COCO train2014 images that appear in none of the training pairs, the development subset, any pair list or any cached shard; an overlap audit finds 0 POPE COCO ids and 0 perceptual-hash matches with GQA testdev images (minimum Hamming distance 50). Every arm passed its registered gates (5,000 rows, identical question sets and un-pruned answers; CDPruner at layer 8 selects the official token set in every row). The design — head, layer 8, 3×3 blocks, budgets and the test rule of Appendix A — was frozen before the run, and the hypotheses were registered before it; with one question per image, the image bootstrap resamples questions. Outside the hypotheses, against CDPruner before the LLM (rescored against our un-pruned answers) SHAPrune keeps more answers at every budget; this does not enter the tally.

N Relation to the closest methods

Table 27 sets SHAPrune beside the methods closest to it in estimator, readout or placement. The entries summarize the cited papers as we read them; only FastV and CDPruner were run here. We did not run LearnPruner, MAP, HAWK or CAPA.

Further related lines. Token reduction predates VLMs: in vision transformers DynamicViT (Rao et al., 2021) trains a light predictor to drop tokens, EViT (Liang et al., 2022) keeps the tokens the class token attends to most and fuses the rest, and ToMe (Bolya et al., 2023) merges similar tokens without training; LLaVA-PruMerge (Shang et al., 2025) selects LLaVA's visual tokens by the vision encoder's class attention and merges the rest before the LLM. VTW (Lin et al., 2025) withdraws all visual tokens after a middle LLM layer, which bears on placement: past some depth the answer needs few or none of them in the sequence. Whether attention weights explain a model's decisions is debated (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019); our readout does not take attention as an explanation but as features from which a learned map predicts removal effects, and it is judged by the answers it keeps. We ran none of these methods.

Table 25: Qwen2.5-VL-7B pruned after layer 8: the full grid of the four registered comparisons, POPE and GQA, *answer recall* and accuracy at 192 / 128 / 64 kept tokens (48 cells). Each cell gives the first arm / the second arm in %, with the verdict for the first arm by the rule of Appendix A in parentheses. POPE rates, like its tests, are on the 5,170 distinct pairs, so the head before the LLM reads 94.68 here against 93.0 on the 9,000 rows of Table 24; GQA uses all 12,578 questions. The head before the LLM is the trained head of Table 24; matched random uses the token sets of Table 24; pure DPP is CDPruner's DPP with constant relevance on Qwen's merger features, and the before-the-LLM row uses the same DPP token sets. †Registered deciding cell (recall at 64 tokens). The 192- and 128-token and accuracy cells were registered as appendix-only; none of the 48 cells enters the 114-cell equal-kept-tokens count of Appendix F, which covers LLaVA only.

	192	128	64
Placement, pure DPP (after layer 8 / before the LLM)			
POPE *answer recall*	98.09 / 98.10 (tie)	97.27 / 97.35 (tie)	95.45 / 95.53 (tie)†
POPE accuracy	91.41 / 91.20 (tie)	90.87 / 90.83 (tie)	89.86 / 89.44 (tie)
GQA *answer recall*	90.01 / 90.04 (tie)	86.21 / 85.78 (tie)	79.92 / 79.44 (tie)†
GQA accuracy	60.42 / 60.33 (tie)	59.67 / 59.53 (tie)	57.56 / 57.58 (tie)
Placement, head (after layer 8 / before the LLM)			
POPE *answer recall*	97.27 / 97.18 (tie)	96.42 / 96.42 (tie)	95.30 / 94.68 (win)†
POPE accuracy	91.88 / 91.93 (tie)	91.49 / 91.49 (tie)	90.33 / 90.06 (tie)
GQA *answer recall*	89.57 / 89.39 (tie)	86.11 / 85.94 (tie)	79.00 / 78.97 (tie)†
GQA accuracy	60.30 / 60.38 (tie)	59.21 / 59.39 (tie)	57.31 / 57.23 (tie)
Head / matched random (both after layer 8)			
POPE *answer recall*	97.27 / 96.87 (tie)	96.42 / 95.71 (tie)	95.30 / 93.60 (win)†
POPE accuracy	91.88 / 90.70 (win)	91.49 / 89.88 (win)	90.33 / 88.09 (win)
GQA *answer recall*	89.57 / 87.57 (win)	86.11 / 84.06 (win)	79.00 / 76.80 (win)†
GQA accuracy	60.30 / 59.52 (tie)	59.21 / 58.40 (tie)	57.31 / 55.45 (win)
Head / pure DPP (both after layer 8)			
POPE *answer recall*	97.27 / 98.09 (loss)	96.42 / 97.27 (loss)	95.30 / 95.45 (tie)†
POPE accuracy	91.88 / 91.41 (tie)	91.49 / 90.87 (tie)	90.33 / 89.86 (tie)
GQA *answer recall*	89.57 / 90.01 (tie)	86.11 / 86.21 (tie)	79.00 / 79.92 (tie)†
GQA accuracy	60.30 / 60.42 (tie)	59.21 / 59.67 (tie)	57.31 / 57.56 (tie)

Table 26: Fresh-image set, 5,000 questions: SHAPRUNE after layer 8 against each comparator, *answer recall* and accuracy at 192 / 128 / 64 tokens. w–ℓ: questions only SHAPRUNE / only the comparator gets right; the 99% interval is SHAPRUNE minus the comparator, in points. H1–H3 were registered with recall at 64 tokens as the deciding cell; H1 and H3 hold and no difference is detected for H2, as registered. The last two blocks are not hypotheses. †The only cell any comparator wins: CDPruner at layer 8 on accuracy at 64 tokens. ‡Run in CDPruner's own environment and scored against our harness's un-pruned answers, since that environment has no un-pruned run; its accuracy uses its own answers against the ground truth. Accuracy is binary exact match with VQAv2's majority answer (`multiple_choice_answer`), not the VQA soft score.

Comparator	k	Metric	SHAPRUNE	Comparator	w–ℓ	p	99% interval
H1: FastV once	192	recall	91.56	89.58	326–227	2.9×10^{-5}	$[+0.78, +3.20]$
	192	acc.	69.96	68.86	211–156	0.0048	$[+0.12, +2.06]$
	128	recall	89.34	83.72	527–246	2.3×10^{-24}	$[+4.20, +7.06]$
	128	acc.	69.60	66.22	354–185	3.0×10^{-13}	$[+2.20, +4.56]$
	64	recall	85.34	71.40	967–270	2.5×10^{-92}	$[+12.20, +15.64]$
	64	acc.	68.38	59.04	711–244	1.5×10^{-53}	$[+7.76, +10.88]$
H2: pure DPP@8	192	recall	91.56	91.94	167–186	0.34	$[-1.34, +0.58]$
	192	acc.	69.96	70.36	110–130	0.22	$[-1.20, +0.42]$
	128	recall	89.34	89.60	234–247	0.58	$[-1.38, +0.86]$
	128	acc.	69.60	69.46	168–161	0.74	$[-0.76, +1.12]$
	64	recall	85.34	84.62	377–341	0.19	$[-0.62, +2.10]$
	64	acc.	68.38	67.96	273–252	0.38	$[-0.76, +1.60]$
H3: matched random@8	192	recall	91.56	85.36	512–202	6.5×10^{-32}	$[+4.84, +7.56]$
	192	acc.	69.96	66.80	325–167	9.4×10^{-13}	$[+2.04, +4.30]$
	128	recall	89.34	81.06	637–223	7.1×10^{-47}	$[+6.80, +9.76]$
	128	acc.	69.60	65.08	427–201	1.1×10^{-19}	$[+3.24, +5.80]$
	64	recall	85.34	73.36	872–273	2.3×10^{-73}	$[+10.30, +13.64]$
	64	acc.	68.38	60.14	651–239	8.7×10^{-45}	$[+6.70, +9.72]$
CDPruner@8	192	recall	91.56	91.90	140–157	0.35	$[-1.28, +0.54]$
	192	acc.	69.96	70.44	82–106	0.093	$[-1.20, +0.24]$
	128	recall	89.34	89.80	183–206	0.26	$[-1.50, +0.56]$
	128	acc.	69.60	69.86	123–136	0.46	$[-1.10, +0.58]$
	64	recall	85.34	86.12	262–301	0.11	$[-2.00, +0.40]$
	64	acc.	68.38	69.48	183–238	0.0084	$[-2.14, -0.06]^{\dagger}$
CDPruner, before LLM‡	192	recall	91.56	88.04	375–199	1.8×10^{-13}	$[+2.28, +4.76]$
	192	acc.	69.96	69.98	171–172	1	$[-0.96, +0.92]$
	128	recall	89.34	85.68	431–248	2.1×10^{-12}	$[+2.32, +4.98]$
	128	acc.	69.60	69.52	229–225	0.89	$[-1.00, +1.20]$
	64	recall	85.34	82.58	462–324	9.7×10^{-7}	$[+1.28, +4.18]$
	64	acc.	68.38	68.56	286–295	0.74	$[-1.40, +1.08]$

Table 27: How SHAPRUNE differs from the closest methods. *Input*: what the selector or attributor reads; *target*: what it is trained to predict; *placement*: where tokens are dropped; *cost*: what inference pays for the selection. Entries for methods we did not run are from their papers.

Method	Input	Training target	Placement	Inference cost
AT2 (Cohen-Wang et al., 2025)	attention of a frozen LLM over its text context, one weight per head	effect of context ablations on the model's own response	text context (attribution; also used to prune it)	attention of all layers over the generated response; pruning then re-generates
vStream (Kang et al., 2026)	text-to-image attention of a frozen VLM, one weight per (layer, head)	log-probability drop when an image region is masked; Pearson objective, as ours	none: post-hoc visual attribution, no pruning	attention of all layers, so not available before the answer
HAWK (Zhu et al., 2026b)	text-to-visual attention from the first LLM layer's query and key projections of the embeddings (RoPE omitted), averaged over text tokens, one static weight per head of that layer	none fitted: each weight is the benchmark-score drop when one head's access to visual tokens is masked, normalized per dataset and averaged over datasets	before the LLM	the first layer's query and key projections on the embeddings
CAPA (Jha & Kim, 2026)	attention times each token's value and output projection, summed over heads	none (training-free; calibration only for its FFN approximation)	during decoding, at selected layers, scored against the last generated token	a contribution norm per token at each generation step
LearnPruner (Takezoe et al., 2026)	vision-encoder token features (MLP), then text-to-visual attention in LLM layer 12	task (next-token) loss with a token-count penalty; VLM frozen	two stages: after the vision encoder, and after LLM layer 12	one MLP pass, a greedy diversity fill, and 12 layers on the stage-1 tokens
MAP (Sun et al., 2026b)	input embeddings; a predictor of middle-layer text-to-visual attention	the middle-layer attention itself, averaged over heads	before the first LLM layer	one predictor forward pass, plus a diversity-based greedy fill
FastV (Chen et al., 2024)	last prompt token's attention in layer K, averaged over heads	none (training-free)	inside the LLM, after layer $K = 2$	K layers at full length
CDPruner (Zhang et al., 2025b)	CLIP question relevance and visual-feature similarity	none (training-free)	before the LLM	greedy DPP, one step per kept token
SHAPRUNE	text-to-image attention of the first 8 LLM layers, 256 head weights	block Shapley values (or single ablations) of the first informative token of the model's own answer	inside the LLM, after layer 8	8 layers at full length; head cost independent of k
SHAPRUNE-D	the same head's block scores as the relevance of CDPruner's DPP over projector features	the same (the DPP has no parameters)	inside the LLM, after layer 8	as SHAPRUNE, plus a greedy DPP, one step per kept token

O SHAPRUNE-D: FULL SETS AND THE POST HOC EQUAL-KEPT-TOKEN TALLY

All statistics in this appendix are post hoc unless marked: they were computed after SHAPRUNE-D was proposed and after its registered comparison with CDPruner@8, from the per-question output files, with the same code path and test rule as the registered tally of SHAPRUNE (Appendix F); run first on SHAPRUNE, that code reproduces the registered tally (83 won, 26 tied, 5 lost of 114) exactly. Only the arm under test changes, SHAPRUNE after layer 8 to SHAPRUNE-D after layer 8; the comparators, benchmarks and budgets are those of the registered axis. Table 28 gives the full-set metrics, Table 29 the tally per comparator, Table 30 every recall cell and Table 31 the first fresh set. On POPE, SHAPRUNE-D's yes-rate is 43.27, 43.87 and 43.58% at 192, 128 and 64 tokens, against 42.22% for the un-pruned model (descriptive). SparseVLM, PyramidDrop and the supervision controls are not in this equal-kept-token tally (SparseVLM enters only the post hoc matched-compute comparison of Appendix G), and SHAPRUNE-D was not run on Qwen2.5-VL-7B.

Table 28: Full sets at every budget after layer 8: SHAPRUNE-D and CDPruner@8 (the rows of Table 3 at 64 tokens). POPE 9,000: recall / accuracy / mean F1; GQA 12,578: recall / accuracy; MME 2,374: recall / perception + cognition; TextVQA 5,000: recall / accuracy. Rates over all rows (POPE not de-duplicated). SHAPRUNE and pure DPP after layer 8: Tables 12 and 22.

Method	k	POPE	GQA	MME	TextVQA
SHAPRUNE-D after layer 8	192	98.22 / 87.07 / 86.15	93.22 / 61.40	96.38 / 1843.2	88.58 / 58.54
SHAPRUNE-D after layer 8	128	97.36 / 87.40 / 86.60	90.97 / 61.33	94.52 / 1848.7	85.48 / 57.98
SHAPRUNE-D after layer 8	64	96.40 / 87.13 / 86.27	86.60 / 60.52	93.30 / 1839.4	80.82 / 57.01
CDPruner@8	192	98.11 / 87.00 / 86.12	93.06 / 61.40	96.12 / 1858.6	87.32 / 57.76
CDPruner@8	128	97.17 / 87.03 / 86.24	90.83 / 61.31	94.90 / 1832.1	84.64 / 57.56
CDPruner@8	64	96.08 / 87.14 / 86.44	86.93 / 60.68	92.38 / 1778.1	79.38 / 56.35

Table 29: Post hoc equal-kept-token tally of SHAPRUNE-D after layer 8 on the 114-cell axis of Appendix F, per comparator: win / tie / loss for SHAPRUNE-D by the rule of §4 (POPE on its 5,170 distinct pairs). Earlier: the comparator acts at its own position (before the LLM, or FastV after layer 2); same depth: after layer 8. Recall total 94 / 19 / 1 (earlier 71 / 1 / 0, same depth 23 / 18 / 1); accuracy total 69 / 45 / 0 (earlier 55 / 17 / 0, same depth 14 / 28 / 0).

Position	Comparator	Recall cells	Recall W/T/L	Accuracy cells	Accuracy W/T/L
earlier	matched random (pre-LLM)	12	12/0/0	12	12/0/0
earlier	FastV (layer 2, official)	12	11/1/0	12	9/3/0
earlier	VisionZip	12	12/0/0	12	9/3/0
earlier	CDPruner (pre-LLM)	12	12/0/0	12	8/4/0
earlier	pure DPP (pre-LLM)	12	12/0/0	12	8/4/0
earlier	DivPrune (pre-LLM)	12	12/0/0	12	9/3/0
same-depth	FastV@8 (official K=8)	12	9/2/1	12	7/5/0
same-depth	pure DPP@8	12	6/6/0	12	1/11/0
same-depth	CDPruner@8	12	2/10/0	12	0/12/0
same-depth	matched random@8	6	6/0/0	6	6/0/0

Table 30: Every recall cell of the post hoc tally of Table 29: SHAPRUNE-D after layer 8 minus the comparator, in points, with the 99% image-level bootstrap interval and the verdict (W win, T tie, L loss for SHAPRUNE-D). POPE on its 5,170 distinct pairs. Exact McNemar p values and discordant counts are in the output of our analysis script, shared with the code.

Comparator	Bench.	192	128	64
Matched random (pre-LLM)	POPE	$+3.40$ [$+2.59$, $+4.24$] W	$+3.81$ [$+2.92$, $+4.76$] W	$+5.73$ [$+4.53$, $+6.90$] W
	GQA	$+9.89$ [$+8.93$, $+10.86$] W	$+11.09$ [$+9.98$, $+12.16$] W	$+12.55$ [$+11.36$, $+13.70$] W
	MME	$+6.82$ [$+5.22$, $+8.47$] W	$+7.54$ [$+5.77$, $+9.39$] W	$+10.91$ [$+8.89$, $+12.93$] W
	TextVQA	$+16.68$ [$+14.95$, $+18.41$] W	$+18.38$ [$+16.61$, $+20.22$] W	$+18.86$ [$+17.06$, $+20.69$] W
FastV (layer 2, official)	POPE	$+5.26$ [$+4.30$, $+6.29$] W	$+7.31$ [$+6.21$, $+8.50$] W	$+11.39$ [$+10.15$, $+12.70$] W
	GQA	$+11.20$ [$+10.01$, $+12.42$] W	$+14.02$ [$+12.76$, $+15.35$] W	$+18.82$ [$+17.37$, $+20.30$] W
	MME	$+1.43$ [$+0.08$, $+2.74$] W	$+2.36$ [$+0.84$, $+3.92$] W	$+7.83$ [$+5.86$, $+9.73$] W
	TextVQA	$+0.64$ [-0.64, $+1.93$] T	$+1.48$ [$+0.12$, $+2.89$] W	$+4.42$ [$+2.88$, $+6.02$] W
VisionZip	POPE	$+2.42$ [$+1.73$, $+3.11$] W	$+3.27$ [$+2.42$, $+4.16$] W	$+5.34$ [$+4.30$, $+6.43$] W
	GQA	$+7.62$ [$+6.65$, $+8.65$] W	$+9.64$ [$+8.52$, $+10.80$] W	$+10.97$ [$+9.73$, $+12.23$] W
	MME	$+3.88$ [$+2.44$, $+5.31$] W	$+3.16$ [$+1.60$, $+4.76$] W	$+5.77$ [$+3.96$, $+7.67$] W
	TextVQA	$+6.56$ [$+5.10$, $+8.08$] W	$+5.88$ [$+4.32$, $+7.41$] W	$+7.18$ [$+5.49$, $+8.88$] W
CDPruner (pre-LLM)	POPE	$+2.28$ [$+1.69$, $+2.89$] W	$+2.07$ [$+1.41$, $+2.78$] W	$+1.72$ [$+0.99$, $+2.46$] W
	GQA	$+5.01$ [$+4.20$, $+5.84$] W	$+5.26$ [$+4.41$, $+6.11$] W	$+4.39$ [$+3.40$, $+5.37$] W
	MME	$+4.13$ [$+2.70$, $+5.60$] W	$+4.63$ [$+3.20$, $+6.11$] W	$+6.19$ [$+4.51$, $+7.96$] W
	TextVQA	$+7.34$ [$+5.92$, $+8.81$] W	$+7.54$ [$+6.09$, $+9.01$] W	$+7.74$ [$+6.18$, $+9.36$] W
Pure DPP (pre-LLM)	POPE	$+2.57$ [$+1.90$, $+3.25$] W	$+2.34$ [$+1.66$, $+3.06$] W	$+1.72$ [$+0.98$, $+2.50$] W
	GQA	$+5.96$ [$+5.07$, $+6.86$] W	$+6.20$ [$+5.22$, $+7.19$] W	$+5.76$ [$+4.69$, $+6.89$] W
	MME	$+4.04$ [$+2.70$, $+5.48$] W	$+4.55$ [$+2.95$, $+6.07$] W	$+5.52$ [$+3.83$, $+7.29$] W
	TextVQA	$+7.12$ [$+5.71$, $+8.58$] W	$+7.04$ [$+5.55$, $+8.52$] W	$+6.52$ [$+4.96$, $+8.05$] W
DivPrune (pre-LLM)	POPE	$+2.46$ [$+1.84$, $+3.11$] W	$+2.34$ [$+1.64$, $+3.04$] W	$+2.09$ [$+1.27$, $+2.92$] W
	GQA	$+6.15$ [$+5.26$, $+7.09$] W	$+6.46$ [$+5.49$, $+7.43$] W	$+6.01$ [$+4.92$, $+7.06$] W
	MME	$+4.80$ [$+3.37$, $+6.23$] W	$+4.97$ [$+3.41$, $+6.53$] W	$+6.66$ [$+4.89$, $+8.51$] W
	TextVQA	$+8.48$ [$+7.04$, $+9.99$] W	$+7.78$ [$+6.25$, $+9.33$] W	$+7.48$ [$+5.90$, $+9.04$] W
FastV@8 (official K=8)	POPE	$+2.36$ [$+1.62$, $+3.13$] W	$+4.49$ [$+3.51$, $+5.48$] W	$+8.53$ [$+7.32$, $+9.80$] W
	GQA	$+6.70$ [$+5.67$, $+7.78$] W	$+10.83$ [$+9.64$, $+12.10$] W	$+16.63$ [$+15.18$, $+18.05$] W
	MME	$+1.22$ [-0.04, $+2.49$] T	$+1.77$ [$+0.21$, $+3.33$] W	$+7.88$ [$+5.94$, $+9.90$] W
	TextVQA	-1.24 [-2.46, -0.02] L	-0.30 [-1.62, $+1.00$] T	$+2.62$ [$+1.08$, $+4.16$] W
Pure DPP@8	POPE	$+0.64$ [$+0.16$, $+1.14$] W	$+0.31$ [-0.25, $+0.91$] T	$+1.10$ [$+0.40$, $+1.82$] W
	GQA	$+1.11$ [$+0.43$, $+1.79$] W	$+1.57$ [$+0.81$, $+2.34$] W	$+1.73$ [$+0.85$, $+2.69$] W
	MME	$+0.51$ [-0.38, $+1.39$] T	$+0.51$ [-0.55, $+1.60$] T	$+1.14$ [-0.13, $+2.44$] T
	TextVQA	$+1.10$ [$+0.08$, $+2.21$] W	$+0.66$ [-0.48, $+1.82$] T	$+1.26$ [-0.02, $+2.54$] T
CDPruner@8	POPE	$+0.10$ [-0.27, $+0.47$] T	$+0.17$ [-0.27, $+0.63$] T	$+0.25$ [-0.37, $+0.83$] T
	GQA	$+0.16$ [-0.40, $+0.75$] T	$+0.14$ [-0.52, $+0.81$] T	-0.33 [-1.09, $+0.46$] T
	MME	$+0.25$ [-0.46, $+1.01$] T	-0.38 [-1.43, $+0.63$] T	$+0.93$ [-0.21, $+2.15$] T
	TextVQA	$+1.26$ [$+0.22$, $+2.30$] W	$+0.84$ [-0.28, $+1.96$] T	$+1.44$ [$+0.18$, $+2.71$] W
Matched random@8	POPE	$+2.80$ [$+2.03$, $+3.58$] W	$+3.40$ [$+2.46$, $+4.36$] W	$+5.42$ [$+4.28$, $+6.54$] W
	GQA	$+6.32$ [$+5.36$, $+7.27$] W	$+7.45$ [$+6.46$, $+8.48$] W	$+9.75$ [$+8.62$, $+10.88$] W

Table 31: First fresh-image set (5,000 questions, one per image; Appendix M), SHAPRUNE-D after layer 8 against each comparator after layer 8, post hoc except CDPruner@8 at 64 tokens (the registered confirmation of the TextVQA win, which failed). w–ℓ: questions only SHAPRUNE-D / only the comparator gets right; 99% interval: SHAPRUNE-D minus comparator, in points. FastV-once: FastV's corrected score applied once (Table 21). ^{W}: SHAPRUNE-D wins by the rule of §4; every other cell is a tie.

Comparator	k	Metric	SHAPRUNE-D	Comparator	w–ℓ	p	99% interval
pure DPP@8	192	recall	91.98	91.94	146-144	0.953	$[-0.82, +0.90]$
	192	acc.	70.16	70.36	88-98	0.509	$[-0.92, +0.52]$
	128	recall	90.18	89.60	218-189	0.165	$[-0.46, +1.62]$
	128	acc.	69.92	69.46	153-130	0.191	$[-0.38, +1.36]$
	64	recall	87.16	84.62	356-229	1.7×10^{-7}	$[+1.32, +3.76]^{\text{W}}$
	64	acc.	69.34	67.96	246-177	0.000924	$[+0.30, +2.44]^{\text{W}}$
CDPruner@8	192	recall	91.98	91.90	121-117	0.846	$[-0.74, +0.90]$
	192	acc.	70.16	70.44	67-81	0.285	$[-0.90, +0.36]$
	128	recall	90.18	89.80	171-152	0.317	$[-0.56, +1.30]$
	128	acc.	69.92	69.86	111-108	0.893	$[-0.72, +0.80]$
	64	recall	87.16	86.12	255-203	0.0171	$[-0.04, +2.14]$
	64	acc.	69.34	69.48	165-172	0.744	$[-1.10, +0.80]$
SHAPRUNE@8	192	recall	91.98	91.56	97-76	0.128	$[-0.26, +1.12]$
	192	acc.	70.16	69.96	59-49	0.387	$[-0.34, +0.74]$
	128	recall	90.18	89.34	131-89	0.00558	$[+0.08, +1.60]^{\text{W}}$
	128	acc.	69.92	69.60	83-67	0.221	$[-0.30, +0.96]$
	64	recall	87.16	85.34	213-122	7.56×10^{-7}	$[+0.88, +2.80]^{\text{W}}$
	64	acc.	69.34	68.38	143-95	0.00225	$[+0.18, +1.74]^{\text{W}}$
matched random@8	192	recall	91.98	85.36	510-179	1.07×10^{-37}	$[+5.30, +7.96]^{\text{W}}$
	192	acc.	70.16	66.80	323-155	1.18×10^{-14}	$[+2.26, +4.46]^{\text{W}}$
	128	recall	90.18	81.06	647-191	1.13×10^{-58}	$[+7.64, +10.58]^{\text{W}}$
	128	acc.	69.92	65.08	429-187	7.57×10^{-23}	$[+3.58, +6.08]^{\text{W}}$
	64	recall	87.16	73.36	909-219	2.81×10^{-100}	$[+12.18, +15.48]^{\text{W}}$
	64	acc.	69.34	60.14	663-203	1.19×10^{-57}	$[+7.70, +10.64]^{\text{W}}$
FastV-once@8	192	recall	91.98	89.58	325-205	2.1×10^{-7}	$[+1.20, +3.58]^{\text{W}}$
	192	acc.	70.16	68.86	209-144	0.000638	$[+0.32, +2.28]^{\text{W}}$
	128	recall	90.18	83.72	540-217	1.32×10^{-32}	$[+5.08, +7.88]^{\text{W}}$
	128	acc.	69.92	66.22	355-170	4.85×10^{-16}	$[+2.52, +4.90]^{\text{W}}$
	64	recall	87.16	71.40	990-202	4.19×10^{-125}	$[+14.10, +17.48]^{\text{W}}$
	64	acc.	69.34	59.04	722-207	2.02×10^{-67}	$[+8.78, +11.78]^{\text{W}}$