THE COST OF A NEAR MISS: COMPUTE-MATCHED EVALUATION OF TRAINING-FREE VISUAL CROPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free visual cropping aims to improve fine-grained visual question answering, but an appended crop, and the passes that localize it, also add compute. We count visual tokens per question and compare cropping methods to no-crop baselines given the same tokens. ViCrop's reported gain reproduces, and HiDe's keeps its sign at a smaller size. On LLaVA-1.5, which reads images at a fixed resolution, ViCrop's crop beats centered, random or repeated second views of identical token count by 18.9 to 21.5 percentage points on V*Bench, and two higher-resolution tiles by 13.1. At HiDe's own budget of 16,384 tokens, a baseline allowed all of HiDe's tokens still trails it by 3.4 and 6.5 points on HR-Bench-4K and half of HR-Bench-8K. At low budgets on a backbone that can spend the same tokens on resolution, neither gain survives: ViCrop's once the answering call is matched, HiDe's once its localization is charged. Pooled over 30 Qwen2.5-VL configurations, matching the answering call lowers ViCrop's gain of 1.81 points by 2.65, leaving no gain distinguishable from zero on natural scenes and a loss of 2.03 on documents. Whether cropping pays depends on whether its box finds the target often enough to beat what the baseline gains by spending the same compute on resolution. Annotated-box crops on V*Bench still gain 19.9 to 30.4 points when matched, but at ViCrop's 256-token setting attention boxes hit the target on only 26.7% to 39.3% of questions. In-place magnification adds no tokens, but pasting an enlarged region over the image can overwrite the target when the box narrowly misses it. Displacing annotated boxes on V*Bench reveals a near-miss trough: nearby misses score 7.8 to 10.8 points below distant ones at comparable hit rates. Real attention boxes pay this price too. On the 44% of their misses whose pasted patch still covers the target, pasting costs 12.5 points more than on the rest, a gap that resampling the same boxes does not show. Resampling, which compresses the surroundings instead, shows no trough on any of the three models, yet with detector boxes it is significantly worse than pasting on five of six document configurations. Cropping should therefore be compared at matched compute, and in-place methods should report how their primitive responds to localization errors.

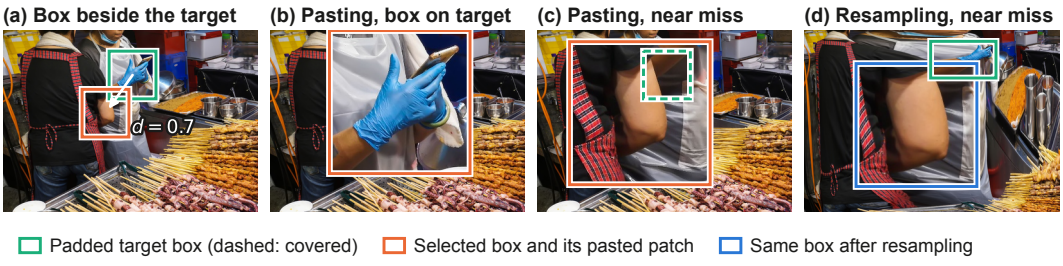

Figure 1: Pasting and resampling on one V*Bench image ($k = 3$, equal visual tokens). (a) The padded annotated box and the same box moved 0.7 of its diagonals away. (b) Pasting a box on the target enlarges it. (c) At the near miss, pasting writes the enlarged patch over the target (dashed). (d) Resampling enlarges the same box by compressing its surroundings; the target survives.

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.

1 INTRODUCTION

A vision-language model sees an image through a fixed budget of visual tokens, and detail finer than that budget can resolve is gone before the language model reads anything (Bai et al., 2025a;b). Training-free visual cropping recovers part of that detail. The model's own attention points to the region a question refers to, that region is cropped and enlarged, and the enlarged view is added to the input (Zhang et al., 2025a). Later methods choose the region by tree search over zoomed patches (Shen et al., 2025) or by attending from the question's key tokens and discarding background (Liu et al., 2026), and all of them report large gains on high-resolution benchmarks such as V*Bench (Wu & Xie, 2024) and HR-Bench (Wang et al., 2025).

These gains are measured against a baseline that does less work. An appended crop adds visual tokens to the answering call, and localization adds forward passes the baseline never runs. On V*Bench with Qwen2.5-VL-7B at ViCrop's 256-token operating point, the cropped input carries 1.59× the visual tokens of the uncropped one. With localization charged at that point, ViCrop and HiDe process 3.6× and 20× the baseline's visual tokens, and ZoomEye, in a run below its published accuracy, spends 3.0× the visual tokens of its own tiled no-zoom baseline in 26.8× as many forward passes (Table 1). A reported gain therefore mixes two effects, better crop placement and more computation, and one budget cannot separate them.

We separate the two with a per-sample protocol (Section 3). Visual tokens are counted from each tokenized input, and the no-crop baseline runs at budgets that bracket the method's count from below and above, because token counts are quantized. What the extra tokens buy depends on the backbone. LLaVA-1.5, on which ViCrop was introduced, reads every image at a fixed resolution. There, ViCrop's crop beats centered, random and repeated second views of identical token count by 18.9 to 21.5 points on V*Bench, and two higher-resolution tiles of the same token count by 13.1, so its published gain stands. HiDe's gain, smaller than published, holds at its own budget of 16,384 tokens, where a baseline allowed all of HiDe's tokens still trails it by 3.4 and 6.5 points on HR-Bench-4K and half of HR-Bench-8K, as more resolution no longer helps. On Qwen2.5-VL, which can spend the same tokens on resolution, a higher-resolution baseline absorbs ViCrop's gain at low budgets, a setting its authors did not report. Pooled over 30 configurations, matching the answering call lowers a significant $+1.81$ by 2.65 $[1.33, 3.87]$ points (bootstrapping configurations too), leaving natural scenes indistinguishable from zero and documents at a loss (Section 4). Whether cropping pays therefore depends on whether its box finds the target often enough to beat what the baseline gains by spending the same compute on resolution. The V*Bench runs include an oracle condition that crops the annotated box, and it gains 19.9 to 30.4 points at matched tokens. There the crop is valuable, and it is the box that fails. Six to seven attention-derived crops in ten do not overlap the target at all.

What a wrong box costs depends on how the crop is used. The methods above append the crop as a second image, so a miss leaves the full image intact and wastes the tokens spent on the crop. Attention boxes mostly miss widely, by a median of 5.4 to 5.8 diagonals of the padded target box. Magnifying the box in place instead adds no visual tokens to the answering call (Appendix E), and localizers that reach the target already exist. An off-the-shelf detector's box overlaps it on up to 69.6% of V*Bench questions (Section 5.3). In place, however, a wrong box changes the very image the model reads.

We measure what that change costs by holding an annotated box fixed in size and shape and translating it by controlled distances (Section 6). When the enlarged region is pasted over the image, accuracy does not decline smoothly as the box moves away. It is lowest just beside the target and partly recovers far from it, a failure mode we call the *near-miss trough*. At matched hit rates a distant miss beats a nearby one by 7.82 to 10.75 points per model when paired by question. A box that still overlaps the target on 58.1% of questions costs 10.47 to 14.66 points on every model.

The trough comes from the primitive, not from magnification itself. Pasting covers the pixels around the box, so a box placed beside the target overwrites the object it was meant to enlarge (Figure 1). Resampling instead enlarges the same box by the same factor and compresses its surroundings, in the manner of attention-guided warping (Dalal et al., 2026), and removes the trough on all three models at identical visual tokens. Resampling has a cost of its own. With detector boxes on document benchmarks, where meaning lives in the page layout, it loses to pasting on five of six configurations. The primitive sets what a localization error costs, and neither primitive is best across domains.

Our contributions are the following.

- **A compute-matched protocol, and when cropping pays under it.** The protocol counts visual tokens per sample, brackets the matched budget from both sides, and matches either the answering call or the total with localization. On a fixed-resolution backbone ViCrop's gain comes from where the crop is placed rather than from the tokens it adds. At low budgets on Qwen2.5-VL, which can raise its resolution, matching lowers the pooled gain by 2.65 points, to one indistinguishable from zero. HiDe keeps its gain at its high budget, where more resolution no longer helps.
- **The near-miss trough and its cause.** Displacing boxes shows that hit rate does not set a box's worth, and a matched primitive comparison attributes the trough to pasting.
- **A diagnostic for what localization must achieve.** A break-even hit rate gives the fraction of targets a localizer must hit, at the payoffs measured for a model and setting, for single-box cropping to beat a baseline given the same total compute. An off-the-shelf detector inside the published architecture, its cost not charged, recovers significant gains on three of nine natural-scene configurations and loses on documents.

2 Related Work

Cropping for fine-grained perception. ViCrop derives a saliency map from attention or gradients, crops the indicated region and appends it to the full image (Zhang et al., 2025a). ZoomEye searches a tree of zoomed sub-patches (Shen et al., 2025), HiDe localizes with the attention of the question's key tokens and removes background from the selected regions (Liu et al., 2026), and other training-free variants select the region with selected attention heads (Xie et al., 2026) or internal representations (Zhong et al., 2025). The training-free methods report gains against a no-crop baseline at the same nominal setting. Trained variants learn a zoom policy (Carvalho et al., 2026), a crop router supervised by the model's own loss gaps (Zhu et al., 2026), or a retrieval policy with a crop-cost penalty (Shabtay et al., 2026). All of them read the selected region at a higher resolution than the full image.

Compute in the cropping literature. Visual token pruning evaluates methods at matched token budgets (Chen et al., 2024; Zhang et al., 2025b), because efficiency is what those methods claim. Cropping work has begun to account for cost: GapSight re-reads a region only when a question calls for local evidence (Zhu et al., 2026), and AwaRes penalizes crop cost during training (Shabtay et al., 2026). In our comparison, the baseline instead receives the visual-token budget the method itself realized on each question. Zhao et al. (2026) show that a correct crop is worth less when the model already receives a high-resolution image, a budget dependence we also find.

Magnification primitives. Crop-and-append methods present the enlarged region as a separate image. In-place magnification instead enlarges the region within a single image of unchanged size. Saliency-guided warping does so for detection (Thavamani et al., 2021), AttWarp warps the image by attention (Dalal et al., 2026), and Perception Magnifier magnifies attended regions while decoding (Mao et al., 2026). Pasting the enlarged region over the image is the simplest in-place construction. We pair it with a box-shaped counterpart of AttWarp's resampling, at the same box, magnification and token count, to isolate what each primitive does to the rest of the image.

Evaluating crops and zoom tools. Because a useful crop is deliberately larger than its target, IoU penalizes correct crops, so we score a crop by its intersection over the ground-truth area, following Rodriguez & Rohrbach (2026). Reinforcement-learned zoom tools (Zheng et al., 2026) form a separate line, and Ma et al. (2026) decompose their gains into intrinsic and tool-induced components, separating the model from the tool, whereas we separate a method from its compute.

3 A Compute-Matched Protocol

A cropping method differs from its baseline in more than where it looks. It presents two images instead of one, runs extra forward passes to localize, and, because the budget applies to each image separately, ends up with more visual tokens in the answering call. Each of these can move accuracy, so attributing a difference to localization requires holding the others fixed. Figure 2 summarizes what we compare, and Figure 6 in the appendix gives the token accounting at the released operating point.

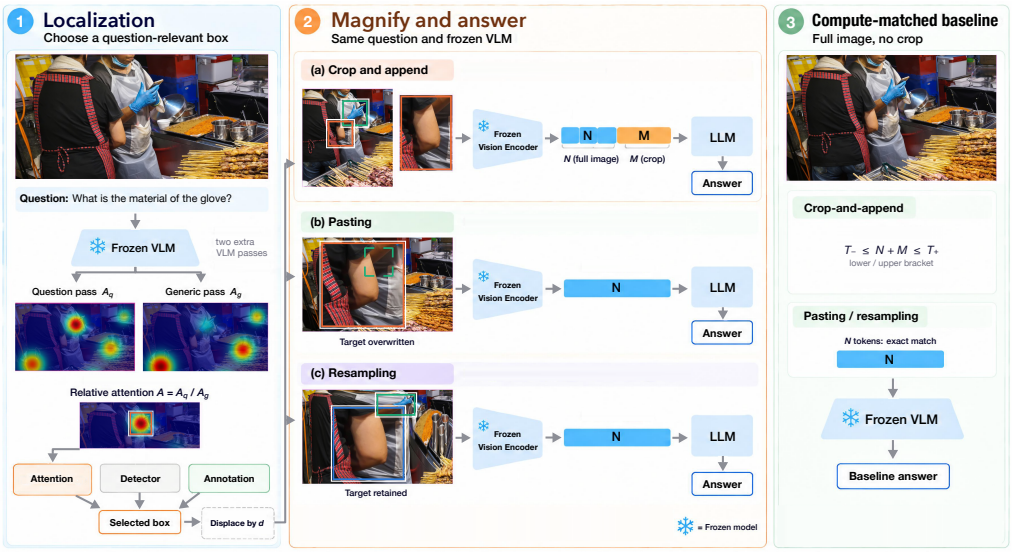

Figure 2: Overview on a V*Bench question. (1) The box comes from the model's relative attention (schematic maps), a detector or the annotation, which the sweep shifts by d. (2) It is magnified by appending a crop, or in place by pasting or resampling; photographs are the actual inputs for a near miss at $d = 0.7$ (green: target, orange: box, blue: resampled box). (3) The baseline reads the full image with visual tokens that bracket the method's for crop-and-append and equal them in place.

3.1 Counting and matching visual tokens

We count visual tokens from the image-token ids in the tokenized input rather than infer them from the configured budget, because the two differ. The budget is an upper bound that interacts with the image aspect ratio and with the processor's rounding to multiples of the patch size. On V*Bench with Qwen2.5-VL-7B at budget 256, the baseline realizes 244.9 tokens and the crop condition 389.8.

For each question we run the cropping condition, read its realized token count T, and run the no-crop baseline at a budget chosen to reach T. Exact equality is usually unattainable, so we run the baseline twice, at realized counts that bracket T, and report the gain at each bracket,

$$T_- \le T \le T_+, \qquad \Delta_\pm = \mathrm{acc}_{\mathrm{crop}} - \mathrm{acc}_{\mathrm{base}}(T_\pm).$$

The ratios T_-/T and T_+/T are 0.92 to 0.99 and 1.03 to 1.11. Forward FLOPs give 0.94 to 1.00 and 1.04 to 1.12 (Appendix A.4). We call this comparison *compute-matched*; in the terms of Section 3.2 it is generation-matched. The text reports the lower bracket Δ_-, which gives the baseline slightly fewer tokens than the method. Table 3 lists both, and we note every conclusion that changes between them.

3.2 Charging localization

Localization costs forward passes that the baseline does not perform. ViCrop's relative attention runs two full passes, one with the question and one with a generic prompt. HiDe runs one at a much larger budget and ZoomEye about a hundred, so we distinguish two levels of matching. A *generation-matched* comparison equalizes only the tokens of the answering call, and a *total-compute-matched* comparison also charges the localization passes and gives the total to the baseline. Table 1 lists the totals. The distinction changes conclusions: at 256 tokens, HiDe's advantage on V*Bench is $+36.65$ points when generation-matched and $+6.81$ when total-compute-matched. ViCrop's re-evaluation in Section 4 is generation-matched, which charges the method less; Section 5.2 also charges localization.

3.3 Setup

We evaluate Qwen2.5-VL-3B and Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B (Bai et al., 2025a) on V*Bench (Wu & Xie, 2024), HR-Bench-4K and HR-Bench-8K (Wang et al., 2025), DocVQA (Mathew et al., 2021) and InfographicVQA (Mathew et al., 2022), at answering budgets

Table 1: Visual tokens processed per question with Qwen2.5-VL-7B at ViCrop's released operating point of 256 visual tokens, including every localization pass, on V*Bench except for the last row. ZoomEye's cost is measured on its own tiling protocol, in a run whose accuracy falls short of its published results (Appendix B.2); its HR-Bench-4K row covers the first 200 items (50 questions) and its ratio uses the no-crop baseline's tokens on those questions. Against its own tiled no-zoom baseline, ZoomEye uses 3.0× the visual tokens and 26.8× the forward passes on V*Bench, and 1.9× the visual tokens on HR-Bench-4K.

	Visual tokens/question	Forward passes	× baseline
No-crop baseline	244.9	1	1.0
ViCrop (generation only)	389.8	1	1.6
ViCrop (with localization)	879.6	3	3.6
HiDe	4,898.7	2	20.0
ZoomEye (V*Bench)	54,604	110.4	223.0
ZoomEye (HR-Bench-4K)	148,557	102.6	585.6

of 256, 512 and 1024 visual tokens. V*Bench has 191 questions, DocVQA and InfographicVQA 800 each, and each HR-Bench version 200 questions posed in four option orders (800 items), which the bootstrap resamples together. The cropping method is ViCrop's relative-attention variant, run at the default layer with its authors' released code, which reproduces their published LLaVA-1.5 gain on V*Bench (Appendix A.3). HiDe and ZoomEye, which need their own software stacks, are evaluated separately (Appendix B). Decoding is greedy, and the two arms of every comparison share one attention kernel, since switching kernels moves accuracy by 2.09 points (Appendix H).

Every comparison is paired over questions. We report 95% intervals from a paired bootstrap with 200,000 resamples and a fixed seed, recompute compute-matched cells within a factor of five of a correction threshold at 2,000,000 resamples, and apply Bonferroni correction over each family of comparisons. All per-question outputs are released, and a checker regenerates the numbers in the paper from them, with the exceptions listed in the reproducibility statement.

4 Re-evaluating Cropping Under Matched Compute

Where the gain comes from. ViCrop was introduced on LLaVA-1.5, which reads every image at 336×336 pixels and can take more visual tokens only as additional images. We keep ViCrop's released pipeline and its 1,152 visual tokens, and replace only the images. On V*Bench, ViCrop's crop (57.59) beats a centered crop, a random crop and a repeat of the full image by 21.47, 19.37 and 18.85 points, and two 336-pixel tiles, which spend the same tokens on resolution, by 13.09 $[+5.76, +20.42]$ (Table 7). On HR-Bench-4K and 8K it beats every such control by 5.75 to 10.38 points (Appendix B.7). On this backbone the gain comes from where the crop is placed, and ViCrop's box overlaps the target on 62.8% of questions (at most 39.3% on Qwen, Section 5.1). Its full cost, 7,636 visual tokens per V*Bench question, exceeds what LLaVA-1.5 accepts in one call, so no single-call baseline can match that total. On Qwen2.5-VL placement helps too, by $+1.91$ $[+1.12, +2.72]$ against a same-token centered crop across 30 configurations, but a higher-resolution baseline absorbs it.

Figure 3 shows the cropping gain in all 45 configurations under both comparisons (both brackets in Table 3). Under the methods' own unmatched comparison, ten configurations show a gain at $p < 0.05$ and four survive Bonferroni correction over the 45 comparisons, while eight show a loss and four of those survive. Under compute matching at the lower bracket, one is positive and eighteen negative at $p < 0.05$, and after Bonferroni correction none is positive and seven are negative. The upper bracket, which gives the baseline slightly more tokens, moves the balance further toward harm, with thirteen configurations negative and none positive after correction. Pooled over the two Qwen2.5-VL models (Table 2), the unmatched gain on natural scenes, $+1.59$ $[+0.42, +2.79]$, becomes -0.05 $[-1.42, +1.33]$ once matched, and $+2.15$ on documents becomes -2.03. Interpolating each question's baseline between the two brackets gives -1.21 $[-2.05, -0.36]$ over the 30 configurations (Appendix B.1). Qwen3-VL-8B, run with Qwen2.5-VL's layer 22, loses unmatched (-3.43) and accounts for five of the seven corrected losses at the lower bracket. With its layer re-selected it still loses on average, -2.29 unmatched and -4.42 matched, although one of fifteen configurations is significantly positive after correction (Appendix B.8).

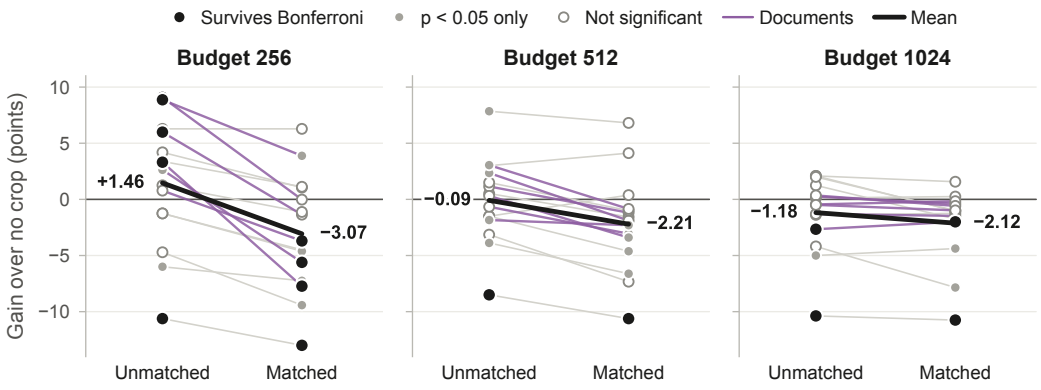

Figure 3: Gains in all 45 configurations, unmatched and matched on the answering call at the lower bracket, one panel per answering budget; each line joins the two gains of one model and benchmark, purple for the two document benchmarks. Black dots survive Bonferroni correction over the 45 comparisons, gray dots are significant at $p < 0.05$ uncorrected, and open dots are not significant. The thick line joins the budget means.

The positive configurations are isolated, and the conclusion is stable to a sixfold change of crop size (Appendices B.1 and B.6). At 256 tokens, once HiDe's localization pass is charged, it keeps a significant advantage in one of eight configurations and reverses to a significant loss in five (Appendix B.3). At HiDe's own setting of up to 16,384 tokens its gain stays positive under total-compute matching and survives correction on HR-Bench-8K only (Appendix B.10). Within its own tiling protocol, ZoomEye leaves no budget at which a compute-matched baseline can be run (Appendix B.2). With 191 questions, V*Bench detects with 80% power only gains above 8.98 points at uncorrected $p < 0.05$, or 13.16 after correction (2.66 to 6.16 uncorrected elsewhere), so a non-significant result there, such as Qwen2.5-VL-7B's $+6.28$ at budget 256, does not rule out a gain. The two brackets agree on the corrected verdict in 39 of 45 configurations and differ by 1.57 points at the median. Quadrupling both budgets shrinks the unmatched gain by 2.64 points on average (Appendix B.4).

Cropping itself is not the problem. The V*Bench runs at budget 256 on all three models, and at 512 on Qwen3-VL-8B, also contain an oracle that crops and appends the annotated box, with its own compute-matched baseline. At matched tokens the oracle gains $+19.90$ on Qwen2.5-VL-3B, $+30.37$ on Qwen2.5-VL-7B, and $+30.37$ and $+25.13$ on Qwen3-VL-8B at budgets 256 and 512. A crop of ViCrop's own size and token count centered on the target beats ViCrop's crop by 25.65 to 29.84 points, so where the box goes, more than its size, sets the oracle's advantage (Appendix B.9). The same accounting that erases the attention-driven gain leaves the oracle strongly positive.

5 Where Localization Fails

5.1 Most attention-derived crops miss the target

V*Bench is the only benchmark in our set with annotated target boxes, so it is where localization can be measured directly. For the box B each model's attention produces and the annotated target G,

$$\mathrm{cov}(B, G) = \frac{|B \cap G|}{|G|}, \qquad \mathrm{hit}(B, G) = \mathbf{1}\left[\mathrm{cov}(B, G) > 0\right].$$

Coverage is the intersection over the ground-truth area (Rodriguez & Rohrbach, 2026) rather than IoU, since a crop is meant to be larger than its object. The box hit rate is 26.7%, 30.4% and 39.3% on Qwen2.5-VL-3B, Qwen2.5-VL-7B and Qwen3-VL-8B at ViCrop's 256-token localization budget. Localizing at 1024 tokens raises the 7B rate to 46.1% (Table 24).

5.2 How good a localizer would have to be

Whether a hit rate suffices depends on what a hit is worth, what a miss costs and what the baseline would gain from the same compute. Let u^+ be the mean accuracy change over the generation-matched baseline on questions where the box hits, and u^- the loss on questions where it misses. Let g be the further gain that baseline obtains, read from its measured accuracy curve, when its budget is raised from the tokens of the answering call to the method's total, which adds the two localization passes. A method with hit rate h beats the total-compute-matched baseline when $hu^+ - (1-h)u^- > g$, which, for $u^+ + u^- > 0$, gives the break-even hit rate

$$h^* = \frac{g + u^-}{u^+ + u^-}.$$

The break-even rate is 54.4% $[28.5, 118.1]$ on Qwen2.5-VL-3B, 45.1% $[25.6, 74.6]$ on Qwen2.5-VL-7B and 95.7% $[71.8, 146.7]$ on Qwen3-VL-8B (layer 22). The measured hit rates are 26.7%, 30.4% and 39.3% (Table 11). Bootstrapped jointly with the hit rate, the shortfall is significant on the 3B model and on Qwen3-VL-8B, but not on the 7B model. Qwen3-VL-8B loses 23.28 points on a missed crop and gains 12.00 on a hit, so a localizer would have to find the target on nearly every question. The payoffs are estimated on the questions attention happens to hit or miss, which may differ in difficulty, and calibrated on one localizer they fail to predict the other on the 3B model (Appendix G.5); h^* is a diagnostic for one model, setting and localizer, not a predictor. Because g charges localization, h^* sets a stricter bar than the generation-matched test of Section 4.

5.3 A better localizer recovers part of the gain on natural scenes

To test whether localization is the bottleneck, we first vary the localizer at an identical token count on V*Bench, magnifying in place by pasting. Box hit rate runs from 26.7% to 39.3% for the models' attention through Grounding DINO (Liu et al., 2024b) at 50.8% and two OWLv2 checkpoints (Minderer et al., 2023) at 67.0% and 69.6%, to 100% for the annotated box. Accuracy tracks hit rate almost linearly on all three models, with correlations between $+0.960$ and $+0.997$.

We then make the same substitution inside the published crop-and-append architecture, where the crop costs tokens and the protocol of Section 3 applies unchanged; only the source of the box differs. With attention boxes, none of fifteen configurations is positive after correction and five are significantly negative. With an open-vocabulary detector (OWLv2-base), three are positive after correction, and on Qwen2.5-VL-7B the V*Bench configuration moves from $+6.28$ to $+15.71$ (Table 15). By domain, the detector is significantly positive on three of nine natural-scene configurations but significantly negative on all six document configurations. Selecting one region assumes that the answer lives in one region, whereas a document question may draw on several parts of the page. Counting the detector's own cost as tokens lowers its in-place V*Bench gain of $+8.38$ to about $+6.33$ (Appendix G.4).

6 The Near-Miss Trough Belongs to the Primitive

The localizer sweep of Section 5.3 suggests that hit rate summarizes what a localizer is worth, but real localizers also differ in box size, shape and error pattern. To vary how far a box misses while holding its size and shape fixed, we displace the annotated box.

6.1 A controlled displacement sweep

We translate the annotated box B, padded by 25% of its size on each side, to

$$B_d = \mathrm{clamp}\left(B + d\,\ell_B\,(\cos\theta, \sin\theta)\right),$$

where ℓ_B is its diagonal, clamping keeps the box inside the image, $d \in \{0, 0.25, 0.4, \ldots, 8\}$ takes thirteen values, and θ is drawn for each question and displacement from a per-question seed, so every model and primitive sees the same boxes. Box size, box shape and the magnification factor k are fixed, and the visual-token count equals the baseline's on every question, so d is the only systematic variable. The curve that results is not monotone (Figure 4, pasting). We call $0.55 \le d \le 1.25$ the *near-miss band*, where the box lies beside the target and the pasted patch, three times its size, reaches it. Accuracy is lowest inside this band, at $d = 0.70$ to 1.00, and recovers as the box moves away.

Three observations show that hit rate cannot account for this shape. Moving the box from $d = 0.25$ to $d = 0.40$ lowers the hit rate from 89.5% to 79.1% and removes a gain of 9.42 to 12.04 points: at $d = 0.40$ no model differs significantly from the unmodified image. At $d = 0.55$ the box still overlaps the target on 58.1% of questions (covering 12.6% of it on average), more often than Grounding DINO or any attention box, and every model loses significantly, by 10.47 to 14.66 points. Across the eight displacements whose hit rate lies between 2.1% and 5.2%, the gain spans 14.13 points on both Qwen2.5-VL models and 17.81 on Qwen3-VL-8B. We compare each of the three near ones ($0.85 \leq d \leq 1.25$) with each of the five far ones ($d \geq 1.5$). Paired by question and averaged over these displacements, far beats near by $+7.82$ $[+3.46, +12.32]$, $+10.75$ $[+6.49, +15.15]$ and $+10.12$ $[+5.76, +14.62]$ on the three models. For boxes that do not overlap the target, far beats near by 8.29 to 10.77 (Appendix D.10). At $k = 2$ and 4 the trough persists and moves with the patch: at $d = 1.5$ the gain is -2.09 to -3.66 at $k = 2$ and -7.85 to -17.80 at $k = 4$ (Appendix D.9).

A box beside the target looks nearly right, yet under pasting that is the worst place to land, so a localizer that moves toward the target gets worse before it gets better. At most one attention miss in ten lands in the band (Appendix D.2), but the patch scales with the localizer's box, and attention boxes exceed their targets. Their patch covers the target on 44.2% of attention misses, where pasting costs 12.46 $[3.12, 21.89]$ points more than on the rest, a gap resampling does not show (Appendix D.12).

6.2 Pasting overwrites the target; resampling does not

The sweep pastes the enlarged region over the image, so a box beside the target covers the target with an enlarged distractor, while a distant box leaves it untouched (Figure 1). The trough should then vanish under resampling, which gives the box the same enlarged area but compresses the surroundings instead of overwriting them, at unchanged image size and identical visual tokens on all 2,483 question–displacement pairs per model. Their linear magnifications agree to within 0.5%.

The trough disappears (Figure 4, resampling). Under pasting, the far box beats the near box in 15 of 15 hit-rate-matched pairs on every model, against 4, 2 and 0 of 15 under resampling (these pairs are not independent). Paired by question, far minus near under resampling is -2.23 $[-5.27, +0.84]$ and -2.51 $[-5.90, +0.87]$ on the two Qwen2.5-VL models and -5.06 $[-8.66, -1.47]$ on Qwen3-VL-8B. The paired advantage of resampling over pasting is concentrated at and just before the near-miss band. On Qwen2.5-VL-7B it is $+10.99$ at $d = 0.40$, reaches $+14.14$ at $d = 0.85$, and is indistinguishable from zero at $d \geq 1.5$. At $d = 0$ resampling gains 1.57 to 4.71 points more than pasting, significant on no model. A plain block of the pasted patch's footprint reproduces the trough, far beating near by 8.52 to 10.86 points, so hiding the target causes the loss (Appendix D.11).

6.3 Neither primitive is best everywhere

Resampling does not simply replace pasting, because it has a failure of its own. We ran both primitives on the same questions, boxes and token counts across all five benchmarks, with boxes from attention, from an off-the-shelf detector and, on V*Bench, from the annotation, for 33 configurations in total with Bonferroni correction over all of them (Figure 5 shows the detector-box cells). With detector boxes, resampling is significantly worse than pasting on five of the six document configurations and averages -8.78 points over all six. On the nine natural-scene configurations it is never significantly worse and averages $+0.76$. A likely reason is that pasting changes a bounded region, whereas resampling moves every pixel and, on a page, can compress away text the model still needs.

With attention boxes, both primitives lose significantly on DocVQA on every model, by 6.52 to 23.11 points, pointing to a wrong region that no primitive repairs. On InfographicVQA the six gains lie between -5.31 and $+2.05$, and only one differs from zero at $p < 0.05$.

6.4 How far the trough generalizes

Lacking annotated boxes beyond V*Bench, we built pseudo-boxes on HR-Bench by keeping the items on which two of three detectors (Grounding DINO and two OWLv2 checkpoints) agree at IoU of at least 0.5 (604 to 608 of 800 items), and ran a fixed 600 on each. Magnifying the pseudo-box at $d = 0$ gains $+11.00$, $+8.67$ and $+5.50$ on the three configurations we ran (Qwen2.5-VL-7B and Qwen3-VL-8B on HR-Bench-4K, Qwen2.5-VL-3B on HR-Bench-8K), so the boxes are informative on average. The direction carries over on the two HR-Bench-4K configurations, most clearly on

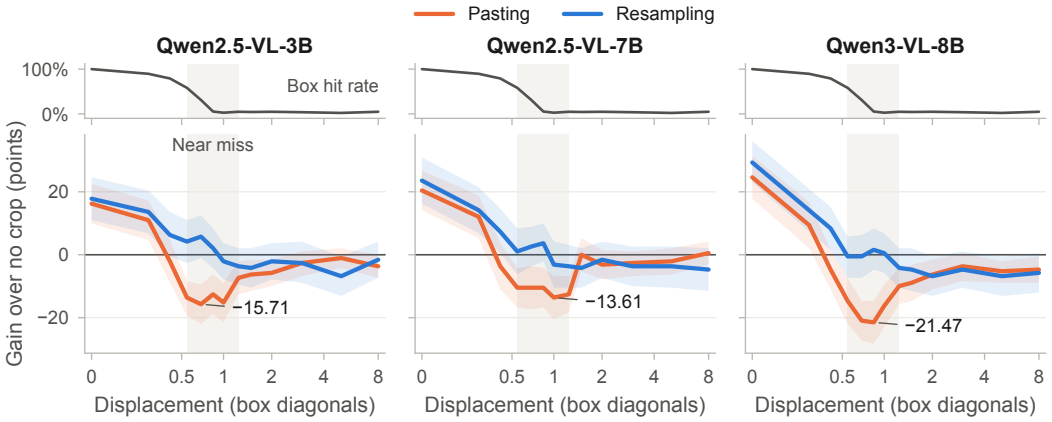

Figure 4: Displacing the annotated V*Bench box by d box diagonals at an identical visual-token count ($n = 191$ per point). Top: box hit rate, identical for every model. Bottom: gain over no crop under pasting and resampling, with 95% bootstrap bands; labels give the deepest point of the pasting curve. Shading marks the near-miss band, $0.55 \leq d \leq 1.25$.

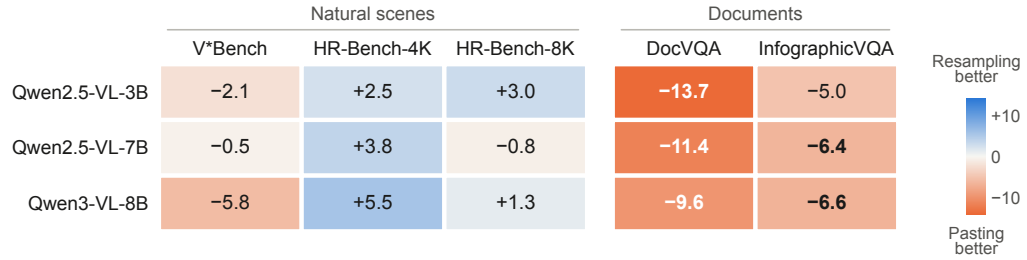

Figure 5: Resampling minus pasting with OWLv2-base boxes, in points. Bold values survive Bonferroni correction over the 33 cells of the full comparison.

Qwen3-VL-8B, where a far miss beats a near one in 11 of 15 pairs under pasting and in 0 of 15 under resampling, but not on HR-Bench-8K, run only with the 3B model. The median far-minus-near difference is much smaller, $+1.17$ and $+1.50$ points against $+8.90$ to $+10.99$ on V*Bench (Table 17).

7 Discussion

Training-free cropping often spends more compute than its published comparisons record. Whether that pays depends on how often the box finds the target, against what the baseline does with the same compute. Placement pays on a fixed-resolution model. On Qwen2.5-VL at low budgets, matching leaves ViCrop no gain distinguishable from zero, but HiDe keeps its gain at 16,384 tokens, where more resolution no longer helps. On V*Bench an annotated box still gains strongly, so the failure lies in the box. At 256 tokens, attention boxes hit the target on 26.7% to 39.3% of questions, against break-even rates of 45.1% to 95.7%, a gap significant on two models. A wrong box costs most under pasting, which covers the target on near misses and on 44.2% of attention misses, a cost set by the primitive. Resampling removes that penalty on natural scenes but, with detector boxes, loses on documents.

Scope. Annotated target boxes exist only on V*Bench, so hit rates and displacements rest on it and on detector-agreement pseudo-boxes, where the trough is weaker. The oracle is a single-box reference, and our resampling is box-shaped rather than the warp of Dalal et al. (2026).

Cropping gains should be reported against a baseline given the same compute, localization included, and in-place methods should report accuracy under displaced boxes.

Reproducibility statement

The artifact accompanies this submission as supplementary material: the per-sample result files for every experiment, the analysis and table-generation scripts, the manuscript sources, and the checker described below.

With one exception noted below, every measured number in this paper is regenerated by scripts in the artifact from the released per-sample files. Two analyses also read the public benchmark images, which we do not redistribute. The one exception to regeneration, a row of Table 24, is marked as not reproducible from the retained files. A checker compares the manuscript with what the scripts produce and fails if any value, multiplicity count or cross-reference disagrees. Packaging refuses to emit an artifact that cannot rebuild its tables byte-for-byte, apart from the two image-dependent tables listed in the artifact's README. Bootstrap intervals use 200,000 resamples from a fixed seed, escalated to 2,000,000 for cells of the main comparisons near a correction threshold. Appendix A.6 describes the regression discipline we applied to the evaluation code itself.

Use of large language models

We used language-model tools as writing and coding aids: they helped draft and edit prose, prepare code and figures, and look up and check related work and citations. The research questions, study design, experimental results and conclusions are our own, and we checked every part of the final manuscript, code and figures.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Miguel Carvalho, Hélder Dias, and Bruno Martins. CropVLM: Learning to zoom for fine-grained vision-language perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 11170–11179, 2026. arXiv:2511.19820.

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.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Dwip Dalal, Gautam Vashishtha, Utkarsh Mishra, Jeonghwan Kim, Madhav Kanda, Hyeonjeong Ha, Svetlana Lazebnik, Heng Ji, and Unnat Jain. Constructive distortion: Improving MLLMs with attention-guided image warping. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.09741.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a. arXiv:2310.03744.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024b. arXiv:2303.05499.

Xianjie Liu, Yiman Hu, Yixiong Zou, Liang Wu, Jian Xu, and Bo Zheng. HiDe: Rethinking the zoom-IN method in high resolution MLLMs via hierarchical decoupling. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2510.00054.

Yan Ma, Weiyu Zhang, Tianle Li, Linge Du, Xuyang Shen, and Pengfei Liu. What does vision tool-use reinforcement learning really learn? Disentangling tool-induced and intrinsic effects for crop-and-zoom. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.01334.

Shunqi Mao, Chaoyi Zhang, and Weidong Cai. Through the magnifying glass: Adaptive perception magnification for hallucination-free VLM decoding. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. arXiv:2503.10183.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. arXiv:2007.00398.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022. arXiv:2104.12756.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.09683.

Jiaxu Qian, Chendong Wang, Yifan Yang, Chaoyun Zhang, et al. Zoomer: Adaptive image focus optimization for black-box MLLM. *Transactions on Machine Learning Research*, 2025. arXiv:2505.00742.

Hector G. Rodriguez and Marcus Rohrbach. SIEVES: Selective prediction generalizes through visual evidence scoring. *arXiv preprint arXiv:2604.25855*, 2026.

Nimrod Shabtay, Moshe Kimhi, Artem Spector, Sivan Haray, Ehud Rivlin, Chaim Baskin, Raja Giryes, and Eli Schwartz. Look where it matters: High-resolution crops retrieval for efficient VLMs. *arXiv preprint arXiv:2603.16932*, 2026.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing multimodal LLMs with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2411.16044.

Chittesh Thavamani, Mengtian Li, Nicolas Cebron, and Deva Ramanan. FOVEA: Foveated image magnification for autonomous navigation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. arXiv:2108.12102.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025. doi: 10.1609/aaai.v39i8.32852.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094, 2024.

Junfei Xie, Peng Pan, and Xulong Zhang. Head-aware visual cropping: Enhancing fine-grained VQA with attention-guided subimage. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026. arXiv:2601.22483.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2502.17422.

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)*, 2025b. arXiv:2410.04417.

Xuanpu Zhao, Zhentao Tan, Dianmo Sheng, Tianxiang Chen, Yao Liu, Yue Wu, Tao Gong, Qi Chu, and Nenghai Yu. Learning to focus and precise cropping: A reinforcement learning framework with information gaps and grounding loss for MLLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2603.27494.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2505.14362.

Liangyu Zhong, Fabio Philipp Rosenthal, Joachim Sicking, Fabian Hüger, Thorsten Bagdonat, Hanno Gottschalk, and Leo Schwinn. FOCUS: Internal MLLM representations for efficient fine-grained visual question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.21710.

Jinchang Zhu, Rong Fu, Yi Ding, Chenghao Wu, Ying Liu, and Menglin Yang. Learning to look again: Loss-gap supervision for free-form crop routing in vision-language models. *arXiv preprint arXiv:2608.21762*, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Protocol Details

The main text defines the compared conditions and the two levels of matching (Section 3); this appendix records what the protocol holds fixed and how significance is computed.

A.1 Resolution mechanisms

Qwen2.5-VL and Qwen3-VL set a continuous pixel cap; InternVL3 selects a discrete number of 448×448 tiles. For tile-based models the bracket must be constructed over tile counts, and the quantization is coarse enough that the bracket above a method's token count overshoots it by a factor of roughly two. We therefore do not report compute-matched results for InternVL3. Appendix B.5 reports the token-matched contrast that its tile quantization does allow. A third class (LLaVA-1.5 (Liu et al., 2024a) and InstructBLIP (Dai et al., 2023)) fixes the visual tokens per image (576 and 32) and admits no budget control, so the protocol cannot construct a bracket. For LLaVA-1.5 we instead match the tokens of the answering call by replacing the appended crop with uninformative second images, or the whole input with tiles (Appendix B.7).

A.2 Benchmarks

V*Bench (Wu & Xie, 2024) (191 questions, the only set in our suite with ground-truth boxes; for the 76 relative-position questions, which name two objects, the annotated box is the union of their two boxes), HR-Bench-4K and HR-Bench-8K (Wang et al., 2025) (800 items each, 200 questions in four option orders), and DocVQA (Mathew et al., 2021) and InfographicVQA (Mathew et al., 2022) (800 validation questions each, drawn at random with seed 0 for DocVQA and at a fixed stride in question-id order for InfographicVQA, scored by ANLS and reported $\times 100$ like accuracy). DocVQA is among the benchmarks ViCrop reports, and InfographicVQA adds a second document domain; Appendix D.7 reports that the detector result reverses on both.

A.3 Fixed across all comparisons

The attention implementation is identical within every comparison. ViCrop and every experiment run on its evaluation harness use `eager`, while the HiDe, ZoomEye and Appendix F harnesses and the confidence-gating attempt of Table 24 use the library default `sdpa` (ZoomEye's code requests FlashAttention, which we replace with `sdpa`). Absolute accuracies are comparable within these sets but not across them; the oracle crop, for instance, reaches 80.10 in the runs of Section 4 and 85.34 in Appendix F. We also fix the dtype (bfloat16), greedy decoding and a limit of 20 new tokens, and for each published method we use the authors' own code, default hyperparameters and default

localization layer, with one exception: ViCrop's crop base size is 336, its LLaVA-1.5 value, whereas its Qwen2.5-VL script uses 224 (Appendix B.6). For ViCrop this is the Qwen2.5-VL implementation its authors released after publication, with attention layer 22, which we apply unchanged to Qwen3-VL-8B. The ViCrop paper reports the method's accuracy only for LLaVA-1.5 and InstructBLIP, so no published number exists on our backbones for our unmatched comparison to reproduce. As a check on the evaluation harness, ViCrop's released LLaVA-1.5 pipeline, run unchanged with our copy of V*Bench, scores 39.79 without cropping and 57.59 with the high-resolution relative-attention variant, a gain of $+17.80$ $[+10.47, +25.13]$ whose interval contains the published $+19.89$ (42.41 and 62.30). We state the attention implementation explicitly because we measured it to move accuracy by 2.09 points (Appendix H).

A.4 Visual tokens and forward FLOPs

The protocol counts visual tokens. To check that token matching is compute matching, we estimate each call's forward FLOPs from the model's architecture and the per-question token counts. The estimate covers the vision encoder, at four patches per visual token (window attention in all but four layers for Qwen2.5-VL, full attention in all 27 layers plus three merger stacks for Qwen3-VL), and the language model's prefill, including the quadratic attention term, at a fixed text length. Decoding a few answer tokens is negligible. Across the 45 configurations the lower-bracket baseline receives 0.94 to 1.00 of the FLOPs of the method's answering call and the upper-bracket baseline 1.04 to 1.12, so the two brackets enclose the method in FLOPs as well as in tokens, and no upper bracket falls below it. The ranges move by at most 0.01 when the assumed text length is halved or increased by half. ViCrop's total cost on V*Bench with Qwen2.5-VL-7B at budget 256, two localization passes plus the answering call, is 3.48 times the no-crop baseline in FLOPs against 3.59 in tokens. Matching only the answering call is also the cheapest accounting any implementation of a crop-and-append method could reach: caching the image prefix can make localization cheaper, but not the answering call itself. Total-compute matching sums visual tokens over a method's calls and gives the sum to one baseline call. In that case, tokens are only a proxy for FLOPs, since one long call has a larger attention term than several short ones while several calls repeat the text; for ViCrop the two agree to within 4% (3.48 against 3.59), and for HiDe and ZoomEye we report tokens only. We did not measure end-to-end latency; Appendix G.4 reports component timings for one detector.

A.5 Statistics

All comparisons are paired at the question level. HR-Bench poses each of its 200 questions in four orders of the answer options (800 items per version); every resample draws questions and keeps a question's four orderings together, so its intervals reflect 200 questions rather than 800 items. Confidence intervals are 95% bootstrap over 200,000 resamples of questions, drawn from a fixed seed (0) so that every interval in this paper is reproducible from the released harness. Intervals are percentiles of the resampled differences, and a p-value is twice the smaller of the fractions of resampled differences at or below zero and at or above zero. The resample count matters at these sample sizes. Accuracy on our smallest benchmark moves in quanta of $1/191 = 0.52$ points, and at 20,000 resamples, a tenth of our default, the Monte-Carlo error on a p-value near 0.003 is about ± 0.0008 (two standard errors, as for every $\pm$ below), which is enough to straddle a Bonferroni threshold of $0.05/15 = 0.0033$. Seven cells of Table 15 fell within a factor of five of that threshold and were recomputed at $2{,}000{,}000$ resamples. The same procedure applies to Table 3, where correcting across forty-five comparisons tightens the threshold to 0.00111. Fourteen compute-matched cells fell within a factor of five of it and were recomputed at $2{,}000{,}000$ resamples, while unmatched cells in that range were not. Throughout, * marks $p < 0.05$ uncorrected and † a result surviving Bonferroni correction over the family named in the caption or text. For the 45 comparisons of the re-evaluation we also apply Benjamini–Hochberg correction and report how many cells survive each correction; other families use Bonferroni only. We do not interpret effects of a single configuration smaller than roughly 1.5 points. We measured cross-architecture numerical divergence of 1.57% of answers on identical inputs (mean per-question disagreement over four conditions, $n = 191$, the same checkpoint on two GPU architectures) and attention-kernel divergence of 8.4% of answers, so differences at that scale are not separable from implementation noise.

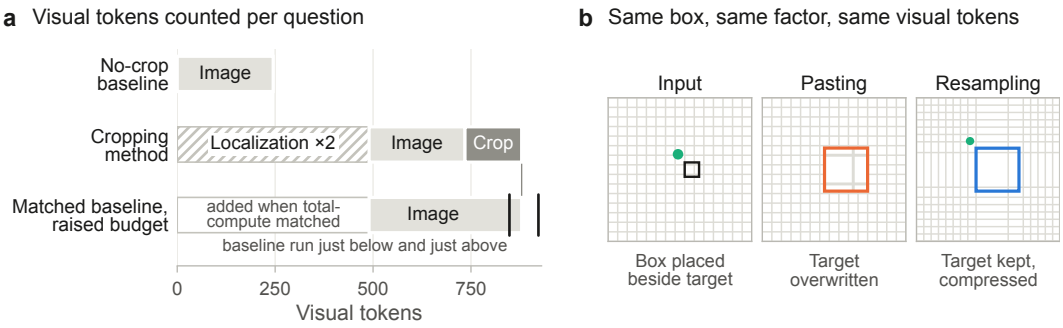

Figure 6: (a) Visual tokens per question on V*Bench with Qwen2.5-VL-7B at the released operating point. The generation-matched baseline receives the tokens of the method's answering call, bracketed by one run just below and one just above; total-compute matching also adds the localization passes. (b) The two primitives on a regular grid, computed with the resampling map used in our experiments: the same box is enlarged by the same factor within an image of unchanged size.

A.6 Regression discipline

Every change to evaluation code was validated by re-running it on questions already computed and requiring exact agreement on all conditions before use. This caught four contaminating changes during the work: a base64 image cache keyed on Python object identity, image pre-shrinking that destroyed small targets, a token-accounting error that omitted 490 localization tokens, and an attention-layer override that silently did not take effect.

B Full Re-Evaluation Results

B.1 Main table

Table 3 lists the 45 configurations summarized in Figure 3; the cropping method in every cell is ViCrop's relative-attention variant. Bonferroni correction over the 45 comparisons sets the threshold at $0.05/45 = 0.00111$. Fourteen compute-matched cells within a factor of five of it were recomputed at 2,000,000 resamples (Appendix A.5). Benjamini–Hochberg at $q = 0.05$ keeps 14 of the 18 cells significant at $p < 0.05$ under the unmatched comparison, 14 of 19 at the lower bracket and 18 of 24 at the upper bracket. The one uncorrected positive at the lower bracket, Qwen3-VL-8B on InfographicVQA at budget 256 ($+3.88$, $p = 0.0079$), does not reproduce at the other budgets (-3.40 and -0.98) and turns significantly negative at 512. The largest matched gains lie on V*Bench with Qwen2.5-VL-7B ($+6.28$ and $+6.81$ at the lower bracket), where power is low (Section 4). Outside V*Bench the largest, Qwen2.5-VL-3B on HR-Bench-4K at budget 512 ($+4.12$ at the lower bracket and $+5.62$ at the upper), is significant only at the upper bracket and only before correction, and does not recur at budgets 256 or 1024 ($+1.12$ and -1.75). Sweeping the base size of the attention crop sixfold, from 168 to 1008, leaves none of seven compute-matched configurations significantly positive, whereas the unmatched comparison moves strongly with crop size in the direction that favors the method (Appendix B.6). For the oracle on Qwen2.5-VL-7B the matched baseline scores below the unmatched one (49.74 against 53.40), since accuracy on 191 questions is not monotone in budget.

Table 2 pools the configurations by model family and domain. Each entry is the unweighted mean of the paired gains in its group. Its interval comes from a paired bootstrap that resamples each benchmark's questions once per replicate and applies the same draw to every configuration on that benchmark, since those configurations answer the same questions (200,000 resamples, seed 0); the configurations themselves are treated as fixed. On natural scenes with the two Qwen2.5-VL models, the lower-bracket interval excludes an average gain above 1.33 points. Resampling the ten model–benchmark units of the two Qwen2.5-VL models as well, with their three budgets together, widens these intervals; with only ten units, these percentile intervals may undercover. The unmatched $+1.81$ stays positive at $[+0.29, +3.55]$. At the lower bracket, documents remain negative at -2.03

$[-3.59, -0.28]$, while the pooled -0.84 widens to $[-2.57, +1.28]$ and natural scenes -0.05 to $[-2.63, +3.03]$, both including zero. The reduction that matching causes stays significant under this interval, at 2.65 $[1.33, 3.87]$ points pooled, 1.64 $[0.13, 3.15]$ on natural scenes and 4.17 $[3.11, 5.21]$ on documents. Under the stricter interval, matching still removes a gain that was there, and it leaves a significant loss only on documents. On Qwen3-VL-8B the method loses under the unmatched comparison and loses more once matched (-3.43 and -5.72 over its fifteen configurations). Against a centered crop of the same token count, ViCrop's crop gains $+1.57$ $[+0.32, +2.83]$ on natural scenes and $+2.42$ $[+1.74, +3.11]$ on documents with the two Qwen2.5-VL models. On Qwen3-VL-8B it gains $+1.90$ $[+0.64, +3.15]$, and $+1.68$ $[-0.34, +3.70]$ on natural scenes alone.

Interpolating instead of choosing a bracket gives the same picture. For each question we take the baseline score at the method's own token count by linear interpolation between the two brackets (the method lies 0.09 to 0.48 of the way from the lower to the upper bracket, averaged per configuration). Pooled over the 30 Qwen2.5-VL configurations the gain is -1.21 $[-2.05, -0.36]$, with -0.17 $[-1.47, +1.14]$ on natural scenes and -2.78 $[-3.59, -1.96]$ on documents; on Qwen3-VL-8B it is -5.91.

Table 2: Cropping gain pooled over configurations, in points: the unweighted mean of the paired gains in each group, with a 95% bootstrap interval that resamples each benchmark's questions jointly across its configurations. Qwen2.5-VL covers the 3B and 7B models. The last column is ViCrop's crop against a centered crop of the same token count, both appended to the full image.

Models	Domain (configs)	Unmatched	Lower bracket	Upper bracket	vs. centered crop
Qwen2.5-VL	Natural (18)	$+1.59$ $[+0.42, +2.79]$	-0.05 $[-1.42, +1.33]$	-0.65 $[-1.99, +0.69]$	$+1.57$ $[+0.32, +2.83]$
Qwen2.5-VL	Documents (12)	$+2.15$ $[+1.45, +2.84]$	-2.03 $[-2.89, -1.16]$	-4.01 $[-4.86, -3.16]$	$+2.42$ $[+1.74, +3.11]$
Qwen2.5-VL	All (30)	$+1.81$ $[+1.06, +2.58]$	-0.84 $[-1.73, +0.06]$	-2.00 $[-2.87, -1.12]$	$+1.91$ $[+1.12, +2.72]$
Qwen3-VL-8B	Natural (9)	-6.27 $[-8.45, -4.10]$	-8.58 $[-10.99, -6.19]$	-9.03 $[-11.51, -6.57]$	$+1.68$ $[-0.34, +3.70]$
Qwen3-VL-8B	Documents (6)	$+0.83$ $[-0.04, +1.70]$	-1.42 $[-2.45, -0.39]$	-1.97 $[-2.99, -0.95]$	$+2.22$ $[+1.36, +3.09]$
Qwen3-VL-8B	All (15)	-3.43 $[-4.79, -2.08]$	-5.72 $[-7.21, -4.23]$	-6.21 $[-7.75, -4.67]$	$+1.90$ $[+0.64, +3.15]$
All	All (45)	$+0.07$ $[-0.64, +0.79]$	-2.47 $[-3.28, -1.64]$	-3.40 $[-4.22, -2.57]$	$+1.91$ $[+1.14, +2.68]$

B.2 ZoomEye under its own protocol

ZoomEye consumes 62× ViCrop's compute on V*Bench, where in our run it gains $+6.81^{*}$ over its own tiled baseline; on HR-Bench-4K it consumes 586× the tokens of the budget-256 no-crop baseline on the same questions and gains $+2.00$ over its tiled baseline (Table 1). Our Qwen2.5-VL-7B run scores 45.03 without search and 51.83 with it on V*Bench, far below what ZoomEye reports for the smaller Qwen2.5-VL-3B (76.96 and 89.01). Rerunning it with the library versions its code pins gives 43.32 and 67.91 on Qwen2.5-VL-3B and 46.52 and 52.41 on Qwen2.5-VL-7B, on the 187 questions that fit in memory under SDPA, so the library version does not explain the gap, and we did not resolve it. We therefore do not interpret ZoomEye's accuracy, and its token and forward-pass counts describe the search as it ran in our environment. The HR-Bench-4K figure covers the first 200 of 800 items (50 questions), because at 102.6 forward passes per question a full run exceeded our compute. ZoomEye's no-zoom baseline is fixed by the tiling implementation at 18,399 tokens on V*Bench and 76,236 on HR-Bench-4K, against which ZoomEye spends 3.0× and 1.9× the visual tokens. That baseline does not respond to `max_pixels` (identical token counts and accuracy at pixel caps of 12.8M, 38.5M and 64.2M), so no compute-matched baseline can be constructed within its protocol.

B.3 HiDe under two levels of matching

Under total-compute matching, HiDe's advantage stays significant at $p < 0.05$ in one of eight configurations ($p = 0.0123$, above the Bonferroni threshold of 0.00625 for eight comparisons), and in five it reverses to a loss that survives correction. The gap between the unmatched number and the total-compute-matched one exceeds 15 points in every configuration. On Qwen2.5-VL-7B and HR-4K, for instance, an unmatched $+21.75^{*}$ becomes $+0.50$, and on Qwen2.5-VL-3B and InfographicVQA an unmatched $+2.31^{*}$ becomes $-26.04^{\dagger}$. HiDe's advantage also disappears on a newer backbone of the same family, where the total-compute-matched baseline with Qwen3-VL-8B on V*Bench reaches 83.77 against HiDe's 84.29. HiDe reports 92.1% on V*Bench with Qwen2.5-VL-7B (Liu

Table 3: All 45 model–benchmark–budget configurations. *Unmatched* is the comparison the methods report, a crop against no crop at the same budget, run by us on the same questions. The compute-matched columns give the no-crop baseline the visual-token count of the method's answering call, from just below (lower bracket) and just above (upper bracket). * marks $p < 0.05$ uncorrected; † marks survival of Bonferroni correction over the 45 comparisons.

					Compute-matched	
Model	Benchmark	Budget	n	Unmatched	Lower bracket	Upper bracket
Qwen2.5-VL-3B	V*Bench	256	191	$+4.19$	$+1.05$	$+1.57$
Qwen2.5-VL-3B	V*Bench	512	191	$+0.52$	-1.57	-3.14
Qwen2.5-VL-3B	V*Bench	1024	191	$+2.09$	-1.05	-1.05
Qwen2.5-VL-3B	HR-4K	256	800	$+3.38^*$	$+1.12$	-1.38
Qwen2.5-VL-3B	HR-4K	512	800	$+3.00^*$	$+4.12$	$+5.62^*$
Qwen2.5-VL-3B	HR-4K	1024	800	$+1.25$	-1.75	-1.50
Qwen2.5-VL-3B	HR-8K	256	800	-1.25	-4.50^*	-5.38^*
Qwen2.5-VL-3B	HR-8K	512	800	-1.50	$+0.38$	-3.88
Qwen2.5-VL-3B	HR-8K	1024	800	$+0.12$	$+0.25$	-3.25
Qwen2.5-VL-3B	DocVQA	256	800	$+3.31^\dagger$	$-7.73^\dagger$	$-10.47^\dagger$
Qwen2.5-VL-3B	DocVQA	512	800	$+1.15$	-1.19	$-3.14^\dagger$
Qwen2.5-VL-3B	DocVQA	1024	800	$+0.24$	-0.38	-0.18
Qwen2.5-VL-3B	InfoVQA	256	800	$+5.99^\dagger$	-1.36	$-5.85^\dagger$
Qwen2.5-VL-3B	InfoVQA	512	800	$+2.35^*$	-1.89	-3.81^*
Qwen2.5-VL-3B	InfoVQA	1024	800	-0.45	-0.16	-3.66^*
Qwen2.5-VL-7B	V*Bench	256	191	$+6.28$	$+6.28$	$+7.85^*$
Qwen2.5-VL-7B	V*Bench	512	191	$+7.85^*$	$+6.81$	$+2.09$
Qwen2.5-VL-7B	V*Bench	1024	191	$+2.09$	$+1.57$	$+1.05$
Qwen2.5-VL-7B	HR-4K	256	800	$+1.25$	-1.12	-3.75
Qwen2.5-VL-7B	HR-4K	512	800	$+1.50$	-1.00	$+1.00$
Qwen2.5-VL-7B	HR-4K	1024	800	$+2.00$	-1.00	-0.62
Qwen2.5-VL-7B	HR-8K	256	800	-1.25	-4.62^*	-1.12
Qwen2.5-VL-7B	HR-8K	512	800	-1.50	-4.62^*	-4.12^*
Qwen2.5-VL-7B	HR-8K	1024	800	-1.38	-1.25	-1.75
Qwen2.5-VL-7B	DocVQA	256	800	$+2.63^*$	$-5.61^\dagger$	$-8.32^\dagger$
Qwen2.5-VL-7B	DocVQA	512	800	-0.66	-3.06^*	$-4.80^\dagger$
Qwen2.5-VL-7B	DocVQA	1024	800	-1.30	-1.47	-1.88^*
Qwen2.5-VL-7B	InfoVQA	256	800	$+9.08^\dagger$	-0.02	-1.92
Qwen2.5-VL-7B	InfoVQA	512	800	$+3.05^*$	-0.85	-3.03^*
Qwen2.5-VL-7B	InfoVQA	1024	800	$+0.35$	-0.58	-1.08
Qwen3-VL-8B	V*Bench	256	191	-4.71	-9.42^*	-8.90^*
Qwen3-VL-8B	V*Bench	512	191	-3.14	-7.33	-7.85^*
Qwen3-VL-8B	V*Bench	1024	191	-4.19	-7.85^*	-6.28
Qwen3-VL-8B	HR-4K	256	800	-6.00^*	-7.25^*	$-7.88^\dagger$
Qwen3-VL-8B	HR-4K	512	800	-3.88^*	-6.62^*	-5.88^*
Qwen3-VL-8B	HR-4K	1024	800	-5.00^*	-4.38^*	$-7.00^\dagger$
Qwen3-VL-8B	HR-8K	256	800	$-10.62^\dagger$	$-13.00^\dagger$	$-15.62^\dagger$
Qwen3-VL-8B	HR-8K	512	800	$-8.50^\dagger$	$-10.62^\dagger$	$-10.50^\dagger$
Qwen3-VL-8B	HR-8K	1024	800	$-10.38^\dagger$	$-10.75^\dagger$	$-11.38^\dagger$
Qwen3-VL-8B	DocVQA	256	800	$+0.78$	$-3.70^\dagger$	$-5.87^\dagger$
Qwen3-VL-8B	DocVQA	512	800	-1.84^*	-2.30^*	$-2.65^\dagger$
Qwen3-VL-8B	DocVQA	1024	800	$-2.66^\dagger$	$-2.00^\dagger$	$-2.21^\dagger$
Qwen3-VL-8B	InfoVQA	256	800	$+8.87^\dagger$	$+3.88^*$	$+1.62$
Qwen3-VL-8B	InfoVQA	512	800	$+0.31$	-3.40^*	-1.76
Qwen3-VL-8B	InfoVQA	1024	800	-0.49	-0.98	-0.94

et al., 2026). Our run of its released code, answering at a budget of 256 visual tokens, reaches 85.86 against a no-crop baseline of 51.31 in the same harness, so our HiDe accuracy sits below the 92.1 it reports. Its own baseline runs at up to 16,384 visual tokens and scores 79.1%, which already brings HiDe's published comparison close to total-compute matching. The Unmatched column here instead gives the baseline the 256-token budget of HiDe's answering call.

Table 4: HiDe's gain under three comparisons, in points, run with its released code. * marks $p < 0.05$ uncorrected; † marks a total-compute-matched result surviving Bonferroni correction over the eight configurations.

Model	Benchmark	Unmatched	Generation-matched	Total-compute-matched
Qwen2.5-VL-3B	HR-4K	$+4.88^{*}$	$+0.00$	$-11.50^{\dagger}$
Qwen2.5-VL-3B	DocVQA	$+1.22$	-12.65^{*}	$-22.31^{\dagger}$
Qwen2.5-VL-3B	InfoVQA	$+2.31^{*}$	-7.41^{*}	$-26.04^{\dagger}$
Qwen2.5-VL-7B	V*Bench	$+34.55^{*}$	$+36.65^{*}$	$+6.81^{*}$
Qwen2.5-VL-7B	HR-4K	$+21.75^{*}$	$+13.88^{*}$	$+0.50$
Qwen2.5-VL-7B	DocVQA	$+9.37^{*}$	-1.51	$-5.83^{\dagger}$
Qwen2.5-VL-7B	InfoVQA	$+16.46^{*}$	$+3.83^{*}$	$-12.03^{\dagger}$
Qwen3-VL-8B	V*Bench	$+26.18^{*}$	$+19.37^{*}$	$+0.52$

B.4 The unmatched gain shrinks as the budget grows

This experiment keeps the unmatched comparison as it is and varies only the visual-token budget given to both systems, so it does not depend on compute matching. Zhao et al. (2026) observed that a ground-truth crop's advantage over a random-noise crop widens as the budget falls, on two trained cropping agents. We extend the observation to the training-free line, to the comparison that published gains report (a crop against no crop), and to paired tests of the change from budget 256 to 1024 (Table 5; Figure 7 shows all three budgets).

Table 5: The unmatched gain at two budgets, with the protocol otherwise unchanged. * marks $p < 0.05$ uncorrected; † marks a change surviving Bonferroni correction over the fifteen configurations.

Model	Benchmark	Gain @256	Gain @1024	Change
Qwen2.5-VL-3B	V*Bench	$+4.19$	$+2.09$	$-2.09\ [-6.81, +2.62]$
Qwen2.5-VL-3B	HR-4K	$+3.38^{*}$	$+1.25$	$-2.12\ [-5.00, +0.75]$
Qwen2.5-VL-3B	HR-8K	-1.25	$+0.12$	$+1.38\ [-1.00, +3.75]$
Qwen2.5-VL-3B	DocVQA	$+3.31^{*}$	$+0.24$	$-3.08\ [-5.00, -1.18]^{\dagger}$
Qwen2.5-VL-3B	InfoVQA	$+5.99^{*}$	-0.45	$-6.44\ [-9.45, -3.44]^{\dagger}$
Qwen2.5-VL-7B	V*Bench	$+6.28$	$+2.09$	$-4.19\ [-9.42, +1.05]$
Qwen2.5-VL-7B	HR-4K	$+1.25$	$+2.00$	$+0.75\ [-2.88, +4.38]$
Qwen2.5-VL-7B	HR-8K	-1.25	-1.38	$-0.12\ [-3.50, +3.38]$
Qwen2.5-VL-7B	DocVQA	$+2.63^{*}$	-1.30	$-3.93\ [-6.34, -1.53]^{\dagger}$
Qwen2.5-VL-7B	InfoVQA	$+9.08^{*}$	$+0.35$	$-8.73\ [-12.00, -5.46]^{\dagger}$
Qwen3-VL-8B	V*Bench	-4.71	-4.19	$+0.52\ [-6.81, +7.85]$
Qwen3-VL-8B	HR-4K	-6.00^{*}	-5.00^{*}	$+1.00\ [-3.00, +5.00]$
Qwen3-VL-8B	HR-8K	-10.62^{*}	-10.38^{*}	$+0.25\ [-3.88, +4.38]$
Qwen3-VL-8B	DocVQA	$+0.78$	-2.66^{*}	$-3.43\ [-5.29, -1.61]^{\dagger}$
Qwen3-VL-8B	InfoVQA	$+8.87^{*}$	-0.49	$-9.36\ [-12.31, -6.44]^{\dagger}$

Ten of fifteen configurations lose gain when the budget is raised fourfold; six of those losses are significant and all six survive correction, and none gains significantly. The mean change is -2.64 points. The decline concentrates on the two document benchmarks, where all six cells fall significantly and five cross from significantly positive to zero or negative. With Qwen2.5-VL-7B on InfographicVQA, for instance, the method reports $+9.08$ at budget 256 and $+0.35$ at 1024. Averaged over the fifteen configurations, the no-crop baseline improves by $+14.12$ points from budget 256 to 1024 and the cropping method by $+11.48$, so the 2.64-point difference is the gain that disappears. The method's token advantage over its baseline falls from $+54\%$ at 256 to $+27\%$ at 512 and $+14\%$ at 1024, but within configurations gain and advantage are not significantly related (ten of fifteen slopes positive, sign test $p = 0.30$), and we do not rely on a correlation pooled across configurations.

B.5 INTERNVL3-8B

InternVL3 selects a discrete number of 448×448 tiles, so no baseline can be run at the method's token count. On V*Bench the method spends 512 visual tokens, and the next attainable tile budget gives the baseline 1,120, $2.2\times$ the method's compute. InternVL3 therefore does not enter Table 3. Tile quantization does permit an exactly token-matched contrast between an attention-derived crop and an uninformative crop at the same visual-token count, which isolates whether the placement of the crop matters.

Table 6: InternVL3-8B, tile-quantized. Values are the accuracy of an attention-derived crop minus that of an uninformative crop of identical token cost. The low tile budget is 512 visual tokens (1,170 on HR-8K), and the high tile budget is 1,343–1,884 depending on benchmark. * marks $p < 0.05$ uncorrected; † marks a change surviving Bonferroni correction over the six tile-budget transitions we measured, one of which (HR-4K from 1,461 to 1,571 tokens) is not shown.

Benchmark	n	Low tile budget	High tile budget	Change
V*Bench	191	$+10.47^*$	-2.62	-13.09 $[-20.42, -5.76]^\dagger$
DocVQA	800	$+6.45^*$	-0.17	-6.62 $[-8.99, -4.31]^\dagger$
InfoVQA	800	$+6.73^*$	-0.56	-7.29 $[-10.50, -4.13]^\dagger$
HR-4K	800	$+0.75$	-2.38^*	-3.12 $[-6.38, -0.12]^*$
HR-8K	800	$+0.88$	-0.50	-1.38 $[-3.38, +0.62]$

At the low tile budget an informative crop is worth up to $+10.47$ points over an uninformative one of the same cost. At the high tile budget that value is gone on all five benchmarks. The decline is significant on four and survives correction on three. The direction replicates Zhao et al. (2026) on an architecture family whose resolution mechanism shares nothing with Qwen's. It shows that an informative crop is worth less as the base budget rises. It cannot show whether cropping loses to a compute-matched baseline on InternVL3, since that baseline could not be built.

B.6 THE CROP SIZE

The result does not rest on the crop size. ViCrop picks its crop adaptively at 1 to 2 times a base size. Every cell of Table 3 uses a base of 336, the value ViCrop's code uses for LLaVA-1.5; its Qwen2.5-VL script uses 224. When we sweep the base from 168 to 1008 (a sixfold range that includes 224, a base we did not run) across V*Bench and DocVQA on two models, none of seven compute-matched configurations is significantly positive and the mean is -1.16. Under the unmatched comparison the same runs move a great deal, in the direction that flatters the method. On DocVQA a 672 crop reports $+8.33^*$ (3B) and $+5.13^*$ (7B), while the same cells under compute matching are -7.14^* and -5.48^*. Enlarging the crop thus raises the reported gain while lowering the compute-matched gain, and the two comparisons disagree about the sign.

B.7 SAME-TOKEN CONTROLS ON LLAVA-1.5

ViCrop was introduced on LLaVA-1.5-7B, which resizes every image to 336×336 pixels (576 visual tokens), so a crop-and-append input carries 1,152. We run ViCrop's released pipeline with its high-resolution relative-attention variant on V*Bench and replace the appended crop with a control, or the whole input with two tiles. The run reproduces the answers of the fidelity check in Appendix A.3 on all 191 questions. Every control keeps the prompt and the token count of ViCrop's input, and all but the tiles keep the full image first; the random crop uses a fixed seed per question, and the two tiles split the image in half along its longer side. ViCrop's box on this backbone intersects the annotated target on 62.8% of the questions, against hit rates of 26.7% to 39.3% for its attention boxes on the three Qwen models (Section 5.1). On LLaVA-1.5 the box comes from the high-resolution variant, which localizes on sub-images, whereas on the Qwen models it comes from a single relative-attention map of the whole image, so the comparison also reflects the difference between the two localization procedures.

On HR-Bench-4K and HR-Bench-8K we first run ViCrop's released LLaVA pipeline to obtain its crops, then apply the same controls with the same harness, which reproduces the pipeline's answers on all 800 items of each. We also add four tiles, a 2×2 split of the image at 2,304 visual tokens.

Table 7: LLaVA-1.5-7B on V*Bench ($n = 191$) with the second image (for the tiles, both images) replaced at an identical visual-token count. The last column is ViCrop's accuracy minus each condition's, with a 95% paired bootstrap interval.

Second image	Visual tokens	Accuracy	ViCrop's crop minus this
none	576	39.79	$+17.80$ $[+10.47, +25.13]$
centered crop, same size	1152	36.13	$+21.47$ $[+14.14, +28.80]$
random crop, same size	1152	38.22	$+19.37$ $[+12.04, +26.70]$
full image repeated	1152	38.74	$+18.85$ $[+11.52, +26.18]$
two tiles instead of the full image	1152	44.50	$+13.09$ $[+5.76, +20.42]$
ViCrop's attention crop	1152	57.59	–
annotated box, 25% padding	1152	73.30	-15.71 $[-23.04, -8.38]$

ViCrop's crop beats every other input on all three benchmarks (Table 8). Seventeen of the eighteen differences are significant at $p < 0.05$ and fifteen survive Bonferroni correction over the eighteen. The exceptions are all on HR-Bench-8K, where the differences from no second image and from two tiles are significant only before correction and the difference from four tiles, at twice the tokens, is not significant ($+4.62$ $[-0.25, +9.62]$, $p = 0.069$). Four tiles score 39.27, 35.00 and 34.12 on the three benchmarks, so tiling does not catch up even with twice the tokens.

Table 8: LLaVA-1.5-7B with ViCrop's released pipeline on V*Bench ($n = 191$) and HR-Bench ($n = 800$ each). Every input keeps ViCrop's prompt; the crops replace the second image, the tiles replace both images, and four tiles use twice the visual tokens. Accuracy, and ViCrop's accuracy minus each condition's with a 95% paired bootstrap interval.

	V*Bench		HR-Bench-4K		HR-Bench-8K	
Input (visual tokens)	Acc.	ViCrop minus this	Acc.	ViCrop minus this	Acc.	ViCrop minus this
none (576)	39.79	$+17.80$ $[+10.47, +25.13]$	35.62	$+8.88$ $[+4.38, +13.50]$	32.50	$+6.25$ $[+1.75, +10.75]$
centered crop, same size (1,152)	36.13	$+21.47$ $[+14.14, +28.80]$	36.38	$+8.12$ $[+3.62, +12.75]$	31.37	$+7.38$ $[+3.00, +11.88]$
random crop, same size (1,152)	38.22	$+19.37$ $[+12.04, +26.70]$	34.12	$+10.38$ $[+6.12, +14.75]$	30.88	$+7.88$ $[+3.50, +12.38]$
full image repeated (1,152)	38.74	$+18.85$ $[+11.52, +26.18]$	35.75	$+8.75$ $[+4.00, +13.63]$	31.62	$+7.12$ $[+2.75, +11.62]$
two tiles (1,152)	44.50	$+13.09$ $[+5.76, +20.42]$	34.38	$+10.12$ $[+5.62, +14.75]$	33.00	$+5.75$ $[+1.38, +10.25]$
four tiles (2,304)	39.27	$+18.32$ $[+11.52, +25.13]$	35.00	$+9.50$ $[+4.75, +14.25]$	34.12	$+4.62$ $[-0.25, +9.62]$
ViCrop's attention crop (1,152)	57.59	–	44.50	–	38.75	–

These comparisons match the answering call only. ViCrop's high-resolution variant localizes on sub-images of at most 1,024 pixels, with two forward passes of 576 visual tokens each, and so spends on average 7,636 visual tokens per question on V*Bench, 18,196 on HR-Bench-4K and 45,850 on HR-Bench-8K, answering call included, against 576 without cropping. On 96.9% of V*Bench questions and on every HR-Bench question this exceeds the 4,096-token context of LLaVA-1.5, so no single-call baseline can be given the same total; the comparison above favors ViCrop in this respect.

B.8 Re-selecting Qwen3-VL-8B's localization layer

ViCrop's released Qwen2.5-VL implementation reads attention from layer 22, which we apply unchanged to Qwen3-VL-8B (Appendix A.3). To test that choice without selecting on accuracy, we measure the hit rate of the relative-attention box at every layer on the first 200 HR-Bench-4K items with detector-agreement pseudo-boxes (Section 6.4), at budget 256 and the default box size; these questions are part of the HR-Bench-4K evaluation, so the re-selection is exploratory rather than a held-out choice. Layer 26 hits 43.5% of the pseudo-boxes and layer 22 hits 38.0%, fifth-best of 36 layers. We rerun all fifteen configurations with layer 26 and nothing else changed (Table 9). At budget 256 the layer matters on V*Bench, where the unmatched gain moves from -4.71 to $+2.09$ and the compute-matched one from -9.42 to -1.57. At budgets 512 and 1024 it moves the V*Bench gains by 2.62 to 4.19 points, and on the other benchmarks by at most 2.26. At budget 256, with either layer, compute matching lowers the gain on every benchmark. Pooled over the fifteen configurations, layer 26 gives -2.29 $[-3.73, -0.86]$ unmatched, -4.42 $[-5.93, -2.93]$ at the lower bracket and -5.15 at the upper. The same three values with layer 22 are -3.43, -5.72 and -6.21. With layer 26, the natural-scene gain goes from -4.45 to -6.57 once matched, and the document gain from $+0.94$ to -1.20. Matching lowers the gain in 13 of the 15 configurations, and one lower-bracket gain

survives correction over the fifteen, InfographicVQA at budget 256 ($+4.87$). The re-selected layer moves Qwen3-VL-8B's pooled gains by about one point and leaves its conclusion unchanged.

Table 9: Qwen3-VL-8B with the released localization layer (22) and with layer 26, re-selected by hit rate on 200 HR-Bench-4K pseudo-boxes: gains in points in all fifteen configurations, unmatched and at the lower and upper brackets, and pooled with the bootstrap of Table 2.

		Layer 22 (released)			Layer 26 (re-selected)		
Benchmark	Budget	Unmatched	Lower	Upper	Unmatched	Lower	Upper
V*Bench	256	-4.71	-9.42	-8.90	$+2.09$	-1.57	-2.62
	512	-3.14	-7.33	-7.85	$+0.52$	-3.66	-4.19
	1024	-4.19	-7.85	-6.28	-0.52	-3.66	-3.66
HR-4K	256	-6.00	-7.25	-7.88	-3.75	-5.00	-5.62
	512	-3.88	-6.62	-5.88	-2.75	-5.50	-4.75
	1024	-5.00	-4.38	-7.00	-4.38	-3.62	-6.25
HR-8K	256	-10.62	-13.00	-15.62	-11.25	-13.63	-16.25
	512	-8.50	-10.62	-10.50	-9.12	-11.25	-11.12
	1024	-10.38	-10.75	-11.38	-10.88	-11.25	-11.88
DocVQA	256	$+0.78$	-3.70	-5.87	$+1.01$	-3.69	-5.58
	512	-1.84	-2.30	-2.65	-1.19	-1.64	-2.02
	1024	-2.66	-2.00	-2.21	-2.45	-1.74	-1.99
InfoVQA	256	$+8.87$	$+3.88$	$+1.62$	$+9.00$	$+4.87$	$+2.00$
	512	$+0.31$	-3.40	-1.76	$+0.20$	-3.26	-1.65
	1024	-0.49	-0.98	-0.94	-0.96	-1.74	-1.60
Pooled	Natural (9)	-6.27	-8.58	-9.03	-4.45	-6.57	-7.37
Pooled	Documents (6)	$+0.83$	-1.42	-1.97	$+0.94$	-1.20	-1.81
Pooled	All (15)	-3.43	-5.72	-6.21	-2.29	-4.42	-5.15

B.9 Position or size of the oracle crop

The oracle crops the annotated box padded by 25% of its size on each side, which is usually smaller than ViCrop's crop, so it differs from ViCrop in both where and how large the crop is. To separate the two we crop a box of exactly ViCrop's size, centered on the annotated target and clamped inside the image, and append it as ViCrop does, on the same V*Bench questions at budget 256. Its visual-token count equals ViCrop's on all 191 questions. Moving the box onto the target at unchanged size gains $+25.65$ $[+18.32, +32.98]$, $+27.75$ $[+19.90, +35.60]$ and $+29.84$ $[+22.51, +37.17]$ points over ViCrop's crop on the three models. Over the lower-bracket baseline it gains $+26.70$, $+34.03$ and $+20.42$. Shrinking it to the padded annotated box changes accuracy by -9.42, -7.33 and $+7.33$, with no consistent sign. Position therefore accounts for most of the oracle's advantage, while the effect of size depends on the model.

B.10 HiDe at its published setting

HiDe reports Qwen2.5-VL-7B with at most 16,384 visual tokens per image for both the baseline and itself: 79.1 and 92.1 on V*Bench, 71.8 and 77.5 on HR-Bench-4K, and 67.9 and 75.4 on HR-Bench-8K (Liu et al., 2026). We run its released code at that setting on the same questions. Our baseline reproduces the published one, 79.06 on V*Bench and 71.62 on HR-Bench-4K, while HiDe reaches 84.82 and 75.50, gains of $+5.76$ $[+1.57, +9.95]$ and $+3.88$ $[+1.00, +7.00]$ against the published $+13.0$ and $+5.7$. We did not resolve the gap between our HiDe accuracy and the published one. The cap decides whether a matched baseline can use the extra tokens. Every V*Bench image fits under it (at most 10,506 tokens), so no baseline can spend more tokens on native resolution and HiDe's gain there stands, as our account predicts; upsampling beyond native resolution, which we did not test, would be a further baseline. On HR-Bench-4K only 32% of the images fit, and no HR-Bench-8K image does, so on both a baseline can still raise its resolution.

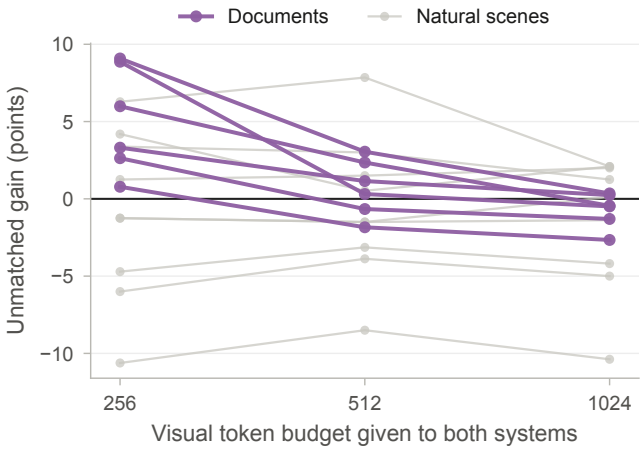

Figure 7: The unmatched gain, one line per model and benchmark, with only the visual-token budget varied for both systems. Purple lines are the two document benchmarks across three models.

On the two HR-Bench sets we give the no-crop baseline the tokens of HiDe's localization pass and answering call together. On HR-Bench-4K the baseline reaches native resolution at 17,971 tokens on average, well below the 30,963 it is offered, and scores 72.12; HiDe still leads it by $+3.38$ $[+0.25, +6.75]$. On the 400 even-numbered HR-Bench-8K items (two option orders of each question) the baseline receives 34,517 tokens on average and reaches 66.75, against 73.25 for HiDe and 67.25 for the capped baseline; HiDe leads it by $+6.50$ $[+1.25, +11.75]$, and a baseline given only the tokens of HiDe's answering call by $+6.75$ $[+1.50, +12.00]$. The extra tokens change the baseline's accuracy by -0.50 on HR-Bench-8K and by $+0.50$ on HR-Bench-4K. At this budget more resolution does little for the baseline, so the baseline gains little from HiDe's extra compute and HiDe's gain survives total-compute matching ($p = 0.040$ on HR-Bench-4K and 0.017 on HR-Bench-8K; with Bonferroni over the two, only HR-Bench-8K stays significant). This is the regime described in Appendix I.2. At the 256 tokens of Section 4, by contrast, the same compute helps the baseline.

C Localization Measurements

C.1 Coverage of attention-derived boxes

We measure a crop by its intersection over the ground-truth area, following Rodriguez & Rohrbach (2026). IoU is unsuitable on V*Bench: targets average 29×81 pixels, so a 336×336 crop that contains the target perfectly still scores an IoU of about 0.02. Beyond the hit rates of Section 5.1, mean coverage of the target is 0.181, 0.265 and 0.327 on Qwen2.5-VL-3B, Qwen2.5-VL-7B and Qwen3-VL-8B.

C.2 Attention concentrates on the target, and the box still misses

ViCrop supports its view that the bottleneck is perception with the observation that attention concentrates on the target, and the observation reproduces. On the 191 V*Bench questions, with Qwen2.5-VL-7B at ViCrop's default layer, relative attention inside the ground-truth box is 8.86 $[6.37, 11.69]$ times the whole-image mean for a box of the same size and exceeds the mean in 62.8% of questions. The signal is question-conditioned: a generic instruction drops the ratio to 0.727 $[0.632, 0.830]$ and the box hit rate to 5.8%, and a shuffled question gives 0.938 and 9.4%. The ratio predicts box success with an AUC of 0.887 (mean ratio 24.16 on hits, 2.19 on misses).

On a third of all questions, attention inside the target is nearly four times the image mean (3.95 on that subset) while the box derived from the same map has zero overlap with it. Concentration shows that the model holds localization information, but not that the information survives the step from an attention map to a box.

Table 10: Attention concentration and box-level hit rate from the same runs: the concentration signal reproduces, and the box derived from it still misses.

Quantity (Qwen2.5-VL-7B, V*Bench, $n = 191$)	Value
Mean attention ratio inside the target	8.86 [6.37, 11.69]
Questions with ratio > 1	62.8%
Questions where the derived box hits the target	30.4%
Questions with ratio > 1 and a box that misses entirely	**35.1%**

C.3 Components of the break-even hit rate

Section 5.2 defines the break-even hit rate. Table 11 gives its components per configuration. The baseline gain g is measured from the accuracy of the generation-matched baseline, which is also the reference for u^+ and u^-, to the point on the baseline's accuracy curve over 11 budgets nearest the method's total. That point lies at 865 tokens on all three models. Appendix G.5 reports the one attempt to use the threshold as a predictor.

Table 11: Break-even hit rate per configuration, measured on V*Bench at budget 256. u^+ and u^- are measured against the generation-matched baseline, and g is that baseline's further gain when given the method's total, including both localization passes.

Model	u^+	u^-	g	h^*	Actual h	Gap
Qwen2.5-VL-3B	+17.65	5.00	+7.33	54.4%	26.7%	−27.7
Qwen2.5-VL-7B	+31.03	4.51	+11.52	45.1%	30.4%	−14.7
Qwen3-VL-8B	+12.00	23.28	+10.47	95.7%	39.3%	−56.4

D Localizer and Primitive Experiments

D.1 Design

Unless stated otherwise, every experiment in this appendix fixes the model, the questions and the magnification, and changes only the localizer or the primitive. The box is magnified at the original image dimensions, so the visual-token count equals the baseline's on every question, verified from `input_ids`. By default the magnified region is pasted over the image, the construction our equal-token experiments use; Section 6.2 compares it with resampling. Pasting resizes the box by k with Lanczos filtering, capping each side at the image's, and pastes the patch centered on the box and clamped inside the image. Resampling keeps the image size and remaps each axis separately. Along x the sampling density is $1 + a\exp(-(x-c)^2/2\sigma^2)$, with c the box center and σ a quarter of the box width; output column i reads the input at $F^{-1}(i/(W-1))$, where F is the normalized cumulative density, and a is set by bisection so that the box spans $\min(kw, 0.999W)$ output columns. The y axis is treated alike, and pixels are interpolated bilinearly. Because the axes are remapped separately, content that shares rows or columns with the box is stretched along one axis and compressed along the other. We measure the realized linear magnification of either primitive by filling the box with one color on a blank image and dividing the geometric mean of that color's width and height in the output by the box's.

What is equalized is the visual-token count of the answering call. Total compute is not equalized, since the attention condition runs two extra forward passes of the VLM and each detector runs its own network (Table 12). Detector queries are noun phrases extracted from the question by a fixed rule, identical across detectors and not tuned.

D.2 Real localizers miss far

Although the localizer sweep of Section 5.3 is nearly linear in hit rate and the displacement sweep of Section 6.1 is not, the two results are compatible. When a detector misses, its box center sits a median of 4.8 to 6.9 box diagonals from the target, in the same units as the sweep (the diagonal of

Table 12: The five localizers, spanning the achievable range of box hit rate at a visual-token count identical on every question.

Localizer	Source	Cost
Model attention	the model's own relative attention map, ViCrop's default	2 passes
Grounding DINO	open-vocabulary detector, referring-expression trained	0.16 s
OWLv2-large (Minderer et al., 2023)	open-vocabulary detector, 3.3 GB	0.47 s
OWLv2-base	open-vocabulary detector, 0.6 GB	0.19 s
Ground truth	the benchmark's annotation	–

the padded ground-truth box), and 5.6% to 14.3% of misses land in the near-miss band $[0.55, 1.25]$. For attention boxes the medians are 5.4 to 5.8 and 7.5% to 10.0% of misses land in the band. Real localizers therefore operate mostly in the far-miss regime, where the sweep is flat and localizers differ only in how often they hit (Figure 8). Restricting the displacement sweep to that regime reproduces the linear relation ($r \geq 0.968$); restricting it to the near-miss points gives $r = 0.72$ to 0.80. Distance in these units understates how often a miss covers the target, because a real localizer's patch scales with its own box (Appendix D.12).

D.3 Localizers from outside the model

Zhang et al. (2025a) test external croppers (SAM, YOLO and CLIP) and report them weaker than every internal variant (55.17 and 56.06 for relative- and gradient-attention against 49.42, 48.84 and 48.55 on LLaVA-1.5 / TextVQA). None of those three queries an open-vocabulary detector with a phrase from the question. Each proposes regions (SAM segments, YOLO detects a closed set of classes, CLIP searches crops) and ranks them by CLIP similarity to the whole question. Our substitution is an open-vocabulary detector queried with a noun phrase from the question, which reaches a 69.6% hit rate, against 30.4% for Qwen2.5-VL-7B's own attention, and is worth $+8.4$ to $+12.6$ points on V*Bench at an unchanged token count.

D.4 The magnification factor

Unless stated otherwise, all experiments magnify by $k = 3$, the intervention's only free parameter. We swept $k \in \{1.5, 2, 3, 4, 6\}$ against three localizers on three models on V*Bench and two localizers on HR-Bench-4K with Qwen2.5-VL-7B.

Table 13: The magnification factor k swept over five values, at a token count identical to the baseline in every cell. $n = 191$, except 189 for OWLv2-base (the questions on which it returns a box), whose values therefore differ slightly from the 191-question ones elsewhere ($+8.47$ against $+8.38$ at $k = 3$). * marks $p < 0.05$ uncorrected.

Model	Localizer	k=1.5	k=2	k=3	k=4	k=6
Qwen2.5-VL-3B	ground truth	$+4.71$	$+10.99^*$	$+16.23^*$	$+20.42^*$	$+29.32^*$
Qwen2.5-VL-3B	OWLv2-base	$+3.17$	$+7.94^*$	$+8.47^*$	$+8.99^*$	$+10.05^*$
Qwen2.5-VL-3B	attention	-3.14	$+0.00$	-4.19	$+0.52$	-8.90^*
Qwen2.5-VL-7B	ground truth	$+8.38^*$	$+13.09^*$	$+20.42^*$	$+23.56^*$	$+29.32^*$
Qwen2.5-VL-7B	OWLv2-base	$+5.82^*$	$+6.88^*$	$+8.47^*$	$+10.05^*$	$+12.17^*$
Qwen2.5-VL-7B	attention	$+0.52$	$+2.62$	$+1.57$	$+1.57$	$+3.66$
Qwen3-VL-8B	ground truth	$+9.95^*$	$+17.28^*$	$+24.61^*$	$+28.80^*$	$+32.46^*$
Qwen3-VL-8B	OWLv2-base	$+3.70$	$+8.99^*$	$+12.70^*$	$+14.29^*$	$+11.64^*$
Qwen3-VL-8B	attention	-1.05	$+4.19$	-3.66	-3.14	-2.62

On HR-Bench-4K with Qwen2.5-VL-7B ($n = 796$ for OWLv2-base and 800 for attention), OWLv2-base gives $+1.13$, -0.25, $+2.76$, $+5.53$, $+5.15$ and attention $+1.00$, $+1.00$, $+2.62$, $+1.25$, -2.62 across the same values of k, none significant. Pooling both benchmarks, a detector is positive in 19 of

20 cells (significant in 13, never significantly negative) and attention in 11 of 20, with one significant negative and no significant positive. The ground-truth box, available on V*Bench only, is positive in 15 of 15 (significant in 14). The oracle gain rises monotonically with k on every model, from $+4.71$ at the lowest to $+32.46$ at the highest, so $k = 3$ is conservative rather than tuned.

D.5 The detector localizes what the question asks about

We ran four query strategies on the same questions, detector and magnification; the last uses the fixed string `object` and carries no information from the question.

Table 14: Four query strategies for OWLv2-base on V*Bench, token counts identical to the baseline throughout. Hit rate is over all 191 questions, counting a question without a box as a miss. Gains are over the questions on which the query yields a box, whose number is in the second column. The last row is the control that carries no question information.

Query strategy	Boxes	Hit rate	3B	7B	Qwen3-VL-8B
Rule-extracted (used throughout)	189	69.6%	$+8.47^{*}$	$+8.47^{*}$	$+12.70^{*}$
Full question verbatim	173	71.2%	$+12.14^{*}$	$+15.61^{*}$	$+12.14^{*}$
Last content word only	177	60.7%	$+4.52$	$+10.17^{*}$	$+5.65$
`object` (no question information)	164	11.5%	-2.44	-1.22	-0.61

Removing the question from the query collapses the hit rate from 69.6% to 11.5% and the gain to nothing on all three models. Paired against that control, the rule-extracted query is worth $+10.49^{*}$, $+11.11^{*}$ and $+12.96^{*}$. Feeding the question verbatim hits more often on the questions where it returns a box (78.6% against 70.4%), but it returns one on 173 questions rather than 189, so over all 191 it hits about as often as the rule (71.2% against 69.6%). Its gains are measured on those 173 questions and are not directly comparable with the rule's; every other number in the paper uses the rule.

D.6 The substitution inside the published architecture

Table 15 gives the grid summarized in Section 5.3. The protocol of Section 3 runs unchanged, and only the source of the box differs.

Table 15: Generation-matched gains at budget 256 and the lower bracket with the localizer swapped inside the published crop-and-append architecture, where the crop costs tokens. * marks $p < 0.05$ uncorrected. $n = 191$ on V*Bench and 800 on every other cell; † marks survival of Bonferroni correction over the fifteen comparisons.

	Qwen2.5-VL-3B		Qwen2.5-VL-7B		Qwen3-VL-8B	
Benchmark	Att.	Det.	Att.	Det.	Att.	Det.
V*Bench	$+1.05$	$+5.24$	$+6.28$	$+15.71^{\dagger}$	-9.42^{*}	$+5.24$
HR-4K	$+1.12$	$+4.25^{*}$	-1.12	$+6.25^{*}$	$-7.25^{\dagger}$	-1.00
HR-8K	-4.50^{*}	$+7.75^{\dagger}$	-4.62^{*}	$+9.50^{\dagger}$	$-13.00^{\dagger}$	$+3.12$
DocVQA	$-7.73^{\dagger}$	$-6.75^{\dagger}$	$-5.61^{\dagger}$	$-5.42^{\dagger}$	$-3.70^{\dagger}$	$-4.61^{\dagger}$
InfoVQA	-1.36	$-4.28^{\dagger}$	-0.02	$-7.95^{\dagger}$	$+3.88^{*}$	$-8.31^{\dagger}$

Attention is positive in four of fifteen cells, in none after correction, and negative in five after correction, with a mean of -3.07. The detector is positive in eight, in three after correction, and averages $+1.25$. The detector is significantly positive in three of nine natural-scene cells, with a mean of $+6.23$, and significantly negative in all six document cells, with a mean of -6.22. One natural-scene cell stays negative under both localizers, Qwen3-VL-8B on HR-Bench-4K (attention $-7.25^{\dagger}$, detector -1.00), on the backbone with the highest break-even hit rate (Section 5.2).

D.7 The equal-token detector grid by benchmark

Across the equal-token grid (three detectors, three models, five benchmarks, 45 cells), 18 cells are positive and 11 negative at $p < 0.05$; after Bonferroni correction, three are positive and six negative.

Table 16: Detector gain by benchmark across the 45-cell equal-token grid.

Benchmark	Mean detector gain	Cells positive
V*Bench	$+7.74$	9/9
HR-Bench-8K	$+5.11$	9/9
HR-Bench-4K	$+4.15$	8/9
DocVQA	-3.04	1/9
InfographicVQA	-3.99	0/9

Natural-scene benchmarks have 26 of 27 detector cells positive and document benchmarks 1 of 18. All three surviving positives are natural-scene cells and all six surviving negatives are document cells. Which detector wins depends on the benchmark. The model's own attention does the most harm on DocVQA, where it changes accuracy by -15.98, -15.65 and -23.11 on the three models, all surviving correction.

D.8 Boundary conditions and per-cell results

Table 17 gives the pseudo-box sweep of Section 6.4.

Table 17: Displacing detector-agreement pseudo-boxes on HR-Bench (600 items per configuration), gains in points against the unmodified image at identical visual tokens. $d = 0$ magnifies the pseudo-box itself; * marks $p < 0.05$ uncorrected. Each near displacement ($0.55 \leq d \leq 1.25$, five values) is paired with each far one ($d = 1.5$, 2 and 8), giving fifteen pairs; the table counts the pairs in which the far box scores higher and gives the median far-minus-near difference. On V*Bench the same median under pasting is $+8.90$ to $+10.99$.

		Pasting			Resampling		
Model	Benchmark	$d = 0$	Far wins	Median	$d = 0$	Far wins	Median
Qwen2.5-VL-7B	HR-4K	$+11.00^*$	12/15	$+1.17$	$+11.67^*$	8/15	$+0.17$
Qwen3-VL-8B	HR-4K	$+8.67^*$	11/15	$+1.50$	$+8.67^*$	0/15	-3.17
Qwen2.5-VL-3B	HR-8K	$+5.50^*$	5/15	-0.67	$+6.17^*$	6/15	-0.17

Two candidate explanations fail to account for why the trough is shallower on HR-Bench pseudo-boxes (Section 6.4). Patch coverage predicts the opposite ordering, since the pasted patch covers a larger fraction of the canvas on HR-Bench (0.14 median) than on V*Bench (0.02). Target size does not explain it either. When V*Bench is split at the median target area, both halves keep a deep trough, with a depth on the smaller versus the larger half of $+7.29$ versus $+11.58$ on the 3B, $+10.42$ versus $+9.47$ on the 7B, and $+11.46$ versus $+11.58$ on Qwen3-VL. An untested possibility is that a detector's box marks an object while a V*Bench annotation marks the region a question needs.

Table 18: Displacing the ground-truth box by d box diagonals: all thirteen values, gains at identical token counts; * marks $p < 0.05$ uncorrected. Hit rate is a single column because it depends on the displacement, not on the model.

d	Hit	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen3-VL-8B
0 (oracle)	100.0%	$+16.23^*$	$+20.42^*$	$+24.61^*$
0.25	89.5%	$+10.99^*$	$+12.04^*$	$+9.42^*$
0.40	79.1%	-2.09	-3.66	-4.71
0.55	58.1%	-13.61^*	-10.47^*	-14.66^*
0.70	30.9%	-15.71^*	-10.47^*	-20.94^*
0.85	5.2%	-12.57^*	-10.47^*	-21.47^*
1.00	2.6%	-15.18^*	-13.61^*	-16.23^*
1.25	4.7%	-7.33^*	-12.57^*	-9.95^*
1.50	4.2%	-6.28^*	$+0.00$	-8.90^*
2.0	4.7%	-5.76^*	-3.14	-6.28^*
3.0	3.7%	-2.62	-2.62	-3.66
5.0	2.1%	-1.05	-2.09	-5.24^*
8.0	4.7%	-3.66	$+0.52$	-4.71^*

Table 19: The five-localizer sweep behind Figure 8a, on V*Bench, $n = 191$ per cell, gains against the unmodified image at the same token count.

	Qwen2.5-VL-3B		Qwen2.5-VL-7B		Qwen3-VL-8B	
Localizer	Hit	Gain	Hit	Gain	Hit	Gain
Model attention	26.7%	-4.19	30.4%	$+1.57$	39.3%	-3.66
Grounding DINO	50.8%	$+1.57$	50.8%	$+5.24$	50.8%	$+5.76$
OWLv2-large	67.0%	$+7.33^{*}$	67.0%	$+7.33^{*}$	67.0%	$+13.09^{*}$
OWLv2-base	69.6%	$+8.38^{*}$	69.6%	$+8.38^{*}$	69.6%	$+12.57^{*}$
Ground truth	100%	$+16.23^{*}$	100%	$+20.42^{*}$	100%	$+24.61^{*}$
Slope (pts per % hit)	$+0.284$		$+0.265$		$+0.445$	
r	$+0.997$		$+0.960$		$+0.983$	

Table 20: Per-cell results for the three detectors (45-cell grid) and for attention, all at zero additional visual tokens. $n = 191$ on V*Bench, 800 elsewhere. † survives Bonferroni correction over the 45 detector comparisons (for the Attention column, over its own 15); * is $p < 0.05$ uncorrected.

Model	Benchmark	Attention	OWLv2-base	OWLv2-large	G-DINO
Qwen2.5-VL-3B	V*Bench	-4.19	$+8.38^{*}$	$+7.33^{*}$	$+1.57$
Qwen2.5-VL-3B	HR-4K	$+0.38$	$+2.00$	$+6.50^{*}$	$+9.25^{\dagger}$
Qwen2.5-VL-3B	HR-8K	-1.12	$+5.50^{*}$	$+5.38^{*}$	$+6.50^{*}$
Qwen2.5-VL-3B	DocVQA	$-15.98^{\dagger}$	$+0.33$	-2.96^{*}	-1.77
Qwen2.5-VL-3B	InfoVQA	-0.81	-2.14	-1.51	$-6.03^{\dagger}$
Qwen2.5-VL-7B	V*Bench	$+1.57$	$+8.38^{*}$	$+7.33^{*}$	$+5.24$
Qwen2.5-VL-7B	HR-4K	$+2.62$	$+2.75$	$+4.62^{*}$	$+7.00^{*}$
Qwen2.5-VL-7B	HR-8K	$+0.75$	$+4.88^{*}$	$+5.25^{*}$	$+4.88^{*}$
Qwen2.5-VL-7B	DocVQA	$-15.65^{\dagger}$	-1.96	-3.53^{*}	-2.74^{*}
Qwen2.5-VL-7B	InfoVQA	$+2.05$	-1.97	-1.93	$-6.33^{\dagger}$
Qwen3-VL-8B	V*Bench	-3.66	$+12.57^{\dagger}$	$+13.09^{\dagger}$	$+5.76$
Qwen3-VL-8B	HR-4K	-2.00	-3.00	$+3.38$	$+4.88^{*}$
Qwen3-VL-8B	HR-8K	-0.62	$+5.50^{*}$	$+4.12$	$+4.00$
Qwen3-VL-8B	DocVQA	$-23.11^{\dagger}$	$-4.59^{\dagger}$	$-5.05^{\dagger}$	$-5.13^{\dagger}$
Qwen3-VL-8B	InfoVQA	-2.51	-3.80^{*}	-4.09^{*}	$-8.08^{\dagger}$

D.9 The trough across magnification factors

We chose the near-miss band and the near and far sets of Section 6.1 after seeing the pasting curve at $k = 3$; the later runs (other k, occlusion, resampling) reuse them unchanged, and the pseudo-box sweep takes the whole near-miss band as its near set. The displacement sweep of Section 6.1 pastes at $k = 3$. We repeat it at $k = 2$ and $k = 4$ with the same questions, displacements, directions, padding and budget. The trough appears at every k on every model (Table 21). Its deepest point lies at $d = 0.55$ to 0.7 for $k = 2$, 0.7 to 1.0 for $k = 3$ and 0.85 to 1.0 for $k = 4$. The displacement at which accuracy recovers also grows with k. At $d = 1.5$ the gain is -2.09 to -3.66 at $k = 2$ but -7.85 to -17.80 at $k = 4$. A larger patch reaches the target from farther away, and the trough follows that geometry. On Qwen3-VL-8B it also deepens, from -15.71 at $k = 2$ to -26.18 at $k = 4$. The far-minus-near difference, computed on the displacements of Section 6.1 for comparability, is significant at every k except $k = 2$ on Qwen3-VL-8B ($+3.07$ $[-0.07, +6.39]$); at $k = 4$ part of that far set still lies inside the trough, which understates the difference.

D.10 Coverage instead of hit rate

A box counts as a hit when it overlaps the target at all. Coverage, the fraction of the target inside the box, falls from 0.594 at $d = 0.25$ to 0.126 at $d = 0.55$ and stays between 0.014 and 0.025 from $d = 0.85$ on, so most hits in the near-miss band are grazing. The trough does not depend on that definition. If we keep for each question only the displacements at which the box covers none

Table 21: Pasting the annotated V*Bench box displaced by d box diagonals at magnification k ($n = 191$; $k = 3$ is the sweep of Section 6.1). Gains in points against the unmodified image at identical visual tokens; bold marks the deepest point. The last column is the paired far-minus-near difference on the hit-rate-matched displacements of Section 6.1 (near 0.85 to 1.25, far 1.5 and above), with a 95% bootstrap interval.

Model	k	d=0	d=0.4	d=0.55	d=0.7	d=0.85	d=1	d=1.25	d=1.5	d=2	d=3	d=8	Far minus near
Qwen2.5-VL-3B	2	+10.99	−3.14	**−13.61**	−12.04	−5.76	−8.38	−4.19	−3.66	−1.05	+0.52	−0.52	+5.06 [+2.23, +8.13]
	3	+16.23	−2.09	−13.61	**−15.71**	−12.57	−15.18	−7.33	−6.28	−5.76	−2.62	−3.66	+7.82 [+3.46, +12.32]
	4	+20.42	−0.52	−8.38	−13.09	−12.04	**−17.80**	−16.23	−12.04	−8.38	−3.14	−3.14	+9.28 [+4.96, +13.68]
Qwen2.5-VL-7B	2	+13.09	−1.57	−7.33	**−7.85**	−6.28	−2.09	−6.28	−2.09	−3.66	+1.05	+0.52	+4.26 [+1.26, +7.33]
	3	+20.42	−3.66	−10.47	−10.47	−10.47	**−13.61**	−12.57	+0.00	−3.14	−2.62	+0.52	+10.75 [+6.49, +15.15]
	4	+23.56	−0.52	−7.33	−8.38	−10.47	**−13.09**	−10.99	−7.85	−5.76	−0.52	−0.52	+8.59 [+4.29, +12.95]
Qwen3-VL-8B	2	+17.28	−1.57	**−15.71**	−12.57	−5.76	−3.66	−5.76	−2.62	−3.14	−0.52	−2.09	+3.07 [−0.07, +6.39]
	3	+24.61	−4.71	−14.66	−20.94	**−21.47**	−16.23	−9.95	−8.90	−6.28	−3.66	−4.71	+10.12 [+5.76, +14.62]
	4	+28.80	−6.81	−13.61	−19.37	**−26.18**	−24.08	−19.90	−17.80	−12.04	−8.90	−3.14	+13.75 [+9.25, +18.39]

of the target, a far miss ($d \geq 1.5$) beats a near one ($0.85 \leq d \leq 1.25$) by $+8.29$ $[+3.96, +12.80]$, $+10.77$ $[+6.53, +15.18]$ and $+10.11$ $[+5.70, +14.68]$ points on the three models, paired by question ($n = 190$). The direction is drawn anew at each displacement, and a box that would cross the image border is clamped inside it at unchanged size, so a distant box can land back on a target near the border. This is why coverage and hit rate rise slightly at the largest displacements.

D.11 Occlusion without magnification

To separate hiding the target from showing an enlarged distractor, we cover exactly the footprint of the pasted patch ($k = 3$) with a block of the image's mean color, at the same displacements, directions and budget as the sweep of Section 6.1, on all 191 V*Bench questions. The block reproduces the trough. Far misses ($d \geq 1.5$) beat near ones ($0.85 \leq d \leq 1.25$) by $+10.86$, $+8.52$ and $+9.63$ points on the three models, against $+7.82$, $+10.75$ and $+10.12$ under pasting. At the near displacements the block changes accuracy against the unmodified image by -14.83, -12.39 and -18.32, and pasting changes it by -11.69, -12.22 and -15.88. The resulting advantage of pasting over the block, $+3.14$, $+0.17$ and $+2.44$, differs from zero only on Qwen2.5-VL-3B. The near-miss loss is therefore the cost of covering the target, and the enlarged content, if anything, softens it.

D.12 Occlusion by real localizers' patches

The near-miss band of Section 6.1 is measured in diagonals of the padded ground-truth box, because the swept box has the target's size. A real localizer's patch is k times its own box, so a miss that is far in those units can still cover the target. For every V*Bench question on which a real box misses the target, we place the pasted patch exactly as the in-place runs do ($k = 3$, centered on the box and clamped to the image) and check whether it overlaps the target. It does on 42.9%, 45.1% and 44.8% of attention misses on the three models, and on 36.0%, 25.0% and 32.7% of Grounding DINO, OWLv2-base and OWLv2-large misses. On attention misses whose patch covers the target, pasting changes accuracy against the unmodified image by -13.33, -11.67 and -25.00 points, compared with -5.00, -1.37 and -6.25 on misses whose patch clears it. The differences, -8.33, -10.30 and -18.75, are individually significant only on Qwen3-VL-8B, and their average is -12.46 $[-21.89, -3.12]$ ($p = 0.009$, bootstrap over questions that keeps each question's three models together). The two groups are not assigned at random, so we check that they are comparable. Pasting the annotated box instead gains about as much on both, $+20.00$ against $+17.50$, $+21.67$ against $+20.55$ and $+25.00$ against $+28.12$, so the questions in the two groups profit about equally from a correct box. Among questions the unmodified image answers correctly, pasting keeps 65.5%, 62.5% and 43.8% of them right when the patch covers the target, against 89.1%, 88.6% and 82.8% when it clears it, so the loss does not come from a difference in baseline accuracy. Resampling the same boxes, which compresses the surroundings but covers nothing, shows no gap between the groups ($+2.16$ $[-7.76, +12.01]$). The advantage of resampling over pasting is therefore 14.62 $[3.68, 25.64]$ points larger on the misses whose patch covers the target than on the rest ($p = 0.009$). On detector misses, which are fewer, the covered-minus-clear difference under pasting is negative in six of nine model–detector cells and significant in one, -40.48 $[-69.05, -14.29]$ for OWLv2-base on Qwen2.5-VL-7B. The occlusion behind the trough under controlled displacement thus also acts on the boxes that current localizers produce.

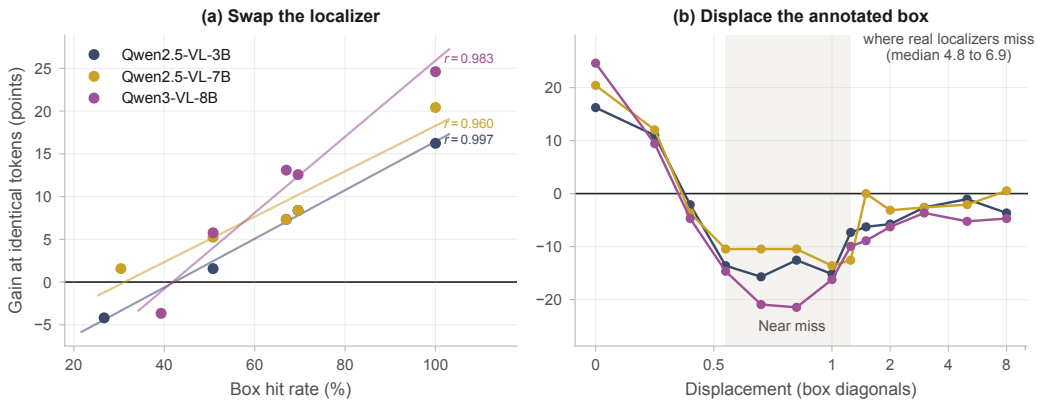

Figure 8: (a) Swapping the localizer at an identical token count: the gain tracks the box hit rate. (b) Displacing the annotated box: the gain is non-monotone, lowest in the shaded near-miss band ($0.55 \leq d \leq 1.25$ box diagonals) and closer to zero beyond 2.

E In-Place Magnification at Identical Visual Tokens

Magnifying the question-relevant region in place, by pasting the enlarged region over the image (labeled foveated below), keeps the image dimensions unchanged, so every condition below sends the model one image with the same number of visual tokens. On V*Bench with Qwen2.5-VL-7B the token counts are identical on each of the 191 questions, so the answering calls need no compute matching; only the attention condition adds forward passes, to localize.

Table 22: The equal-token comparison on V*Bench with Qwen2.5-VL-7B ($n = 191$). All four conditions send the model one image of the same dimensions and the same visual-token count.

Condition	Accuracy	Visual tokens	vs. baseline
Unmodified image	53.40	245	–
Foveated at the ground-truth region	**73.82**	245	$+20.42$ $[+14.14, +26.70]^*$
Foveated at the image center	49.21	245	-4.19 $[-9.95, +1.57]$
Foveated at the attention-predicted region	54.97	245	$+1.57$ $[-5.24, +8.38]$

Knowing where to magnify is worth $+20.42$ points (Figure 9) at no extra tokens or forward passes. Magnifying the wrong place does not help: the center-foveated condition loses 4.19 points, a change not distinguishable from zero. Attention boxes, with a hit rate of 30.4%, deliver $+1.57$, about 8% of what the annotated box delivers. The annotated-box condition needs ground-truth boxes and can therefore be measured on V*Bench only.

F What a Correct Crop Provides

We separate three accounts of what an oracle crop supplies, on the same 191 V*Bench questions within a single run of Qwen2.5-VL-7B. In this run the unmodified baseline scores 51.3% and the oracle crop (49 extra visual tokens) scores 85.3%. The baseline differs from the 53.4% of Table 22 by the 2.09 points we trace to the attention kernel (Appendix H): this run uses the library default, so every comparison below is made within the run.

Figure 10 summarizes three contrasts, which overlap and do not add up. A second image of the same size does not reproduce the gain. A randomly placed crop gives $+3.14$ points $[-1.05, +7.33]$, about 9% of the oracle gain, and a centered one $+1.05$, neither distinguishable from zero. Pointing to the target does not reproduce it either. A red rectangle drawn around the ground-truth region, at a token count identical to the baseline's, is worth $+1.57$ $[-4.71, +7.85]$, and -2.62 $[-7.85, +2.62]$ at native resolution. The gain requires the crop to be on the target. Placing the crop correctly rather than randomly is worth $+30.89$ $[+22.51, +39.27]$, and cropping rather than marking the same region is

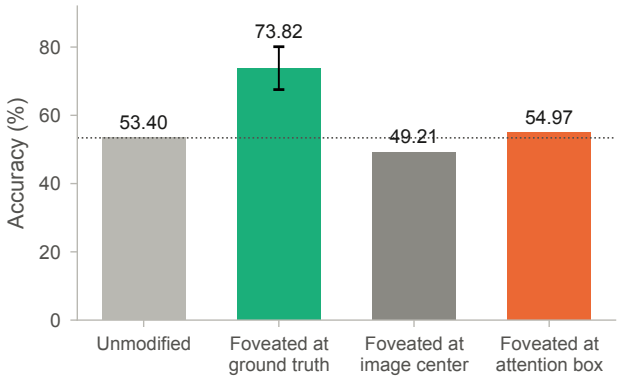

Figure 9: The equal-token comparison. All four conditions send the model one image of the same dimensions, tokenized into the same number of visual tokens on each of the 191 questions (245 on average). The error bar is the 95% bootstrap interval on the ground-truth condition.

Table 23: The seven conditions behind the decomposition, all on the same 191 questions under one model, one attention kernel (SDPA) and, except at native resolution, one budget.

	Condition	Accuracy	Visual tokens
A	Full image	51.31	245
F	Full image + ground-truth crop	**85.34**	294
Fr	Full image + same-size random crop	54.45	294
Fc	Full image + same-size center crop	52.36	294
Mk	Full image, target outlined	52.88	245
N	Full image at native resolution	79.06	4454
MkN	Native resolution, target outlined	76.44	4454

worth $+32.46$ $[+24.08, +40.84]$. The model does not act on a pointer to the region, but it does act on an enlarged copy of it. A crop also separates the region from its background, which HiDe identifies as the main obstacle (Liu et al., 2026), and these contrasts cannot tell enlargement from background removal.

One comparison remains unresolved. The oracle crop at 294 tokens reaches 85.3% while the full image at its native 4454 tokens reaches 79.1%, a difference of $+6.28$ $[+0.00, +12.57]$ whose interval touches zero. The difference shows no linear association with target size ($r = +0.015$), so these data give no support to the account that a small target is diluted among many visual tokens, although they cannot rule it out. We have no alternative explanation.

G Fifteen Attempts

We built fifteen candidate methods, each with a success criterion fixed before the run, and all fifteen fell short. The failures locate where the difficulty lies.

G.1 The criteria

Every attempt was required to beat a total-compute-matched baseline, one also given the attempt's own localization passes, by a margin significant at the 95% level after correction, on at least two benchmarks. Beating the unmatched baseline was not sufficient.

G.2 What was tried

Every value except the marked row was regenerated from the released result files by `scripts/attempts.py` at the resample count and seed of Appendix A.5. Compute-matched

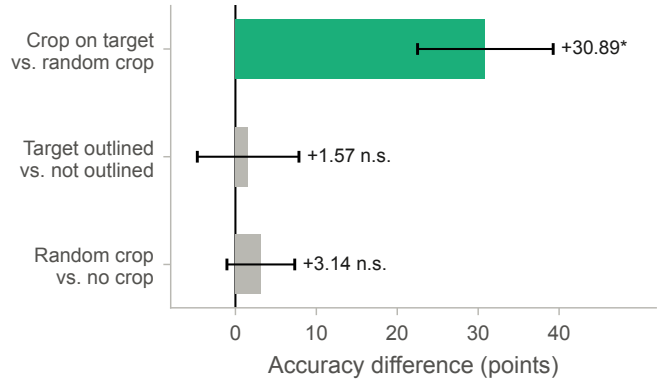

Figure 10: Three contrasts behind the $+34.03$ oracle-crop gain on the same 191 questions. A randomly placed second image and a box drawn around the target are both indistinguishable from zero; only a crop placed on the target carries the gain. The contrasts overlap and do not separate enlargement from background removal.

Table 24: The fifteen attempts, each with its success criterion fixed before the run and the measurement that decided it. The two equal-token foveation rows are gains at identical visual tokens without charging localization; they fail the gate even so.

Attempt	Outcome	Key number (gate: total-compute-matched)
Larger localization budget	coverage saturates	coverage 0.378 at 512, peaks 0.404 at 1024, falls to 0.367 at 2048
Hierarchical low-budget localization	fails gate	$-6.28/-1.57/-3.14$ over three stage settings, none significant
Attention-signal adaptive box size	no headroom	optimistic bound $0.609 <$ always-large 0.611 *(not reproducible from retained files)*
Multi-scale localization union	too expensive	69.6% hit at four scales, 3,840 localization tokens vs ViCrop's 490
Top-k multi-crop	fails gate	$+2.62/-6.88^*/-13.44^*$
Top-k without the full image	fails gate	$-0.52/-17.50^*$
Enlarging the crop	no monotone gain	$+2.62/+6.28/+5.24/+6.28$ at box scales $168/336/672/1008$, unmatched comparison
Confidence gating	underpowered, dropped	$+8.33$ $[0.00, +16.67]$, $p = 0.064$, $n = 60$ **only**
Adaptive localization budget	no headroom	$0.409 \approx$ always-high 0.404, and costlier
Budget reallocation (attention)	fails gate	-5.50^* to -14.35^* across four split points and three configurations
Multi-layer attention fusion	does not transfer	$+13.09^*$ on V*Bench, 1/4 elsewhere
Gradient×attention fusion	does not transfer	$+15.18^*$ on V*Bench, 1/4 elsewhere
Raw (un-normalized) attention	premise false	22.5% hit at budget 1024 vs 46.1% for normalized
Equal-token foveation, attention-located	fails gate	$+1.57/-4.19/-3.66$, none significant
Equal-token foveation, OWLv2-located	fails gate	$+8.38^*/+8.38^*/+2.75/-1.96$, 1/4 after correction

gates use total-compute matching. The marked row could not be reproduced from the retained files and keeps its originally recorded value and status.

G.3 Two attempts that would have looked like progress

Hierarchical localization and top-k multi-crop both beat ViCrop under the unmatched comparison on V*Bench. They gain $+10.99$ $[+4.71, +17.28]$ and $+13.61$ $[+6.81, +20.42]$ against ViCrop's $+6.28$ $[+0.00, +12.57]$, and unlike ViCrop's interval, both of theirs exclude zero. Under total-

compute matching, hierarchical localization lands at -6.28, -1.57 and -3.14 across its three stage settings, none significant, and top-k gives $+2.62$ on V*Bench and significant losses on the other two benchmarks (-6.88^*, -13.44^*). We built both as candidate contributions and found the problem only when we applied our own accounting.

G.4 The attempt that came closest

The last attempt listed replaces the model's attention with OWLv2-base and magnifies in place, so the visual-token count is unchanged on every question (0 of 191 and 0 of 800 questions differ). It is the only attempt with a significant positive result on more than one model.

Table 25: The attempt that came closest: OWLv2-base boxes with in-place magnification, at an unchanged visual-token count.

Model	Benchmark	Box hit rate	Gain over baseline	p
Qwen2.5-VL-7B	V*Bench	69.6%	$+8.38$ $[+2.09, +15.18]$	0.0141
Qwen2.5-VL-3B	V*Bench	69.6%	$+8.38$ $[+2.62, +14.66]$	0.0074
Qwen2.5-VL-7B	HR-Bench-4K	–	$+2.75$ $[-2.50, +8.00]$	0.1087
Qwen2.5-VL-7B	DocVQA	–	-1.96 $[-4.23, +0.30]$	0.0896

It falls short on three counts. First, Bonferroni over the four comparisons requires $p < 0.0125$, and the one cell that clears it leaves a single benchmark where the gate asked for two. Second, the detector is not free. OWLv2-base takes 0.19 s per question in Table 12 and 0.269 s in the Qwen2.5-VL-7B V*Bench run, against 0.283 s for that model's generation at 245 visual tokens. Charging the 0.269 s as 233 visual-token equivalents and reading the baseline off its measured budget curve at 478 tokens (55.45) turns $+8.38$ into $+6.33$, an interpolated point estimate, given without an interval, that assumes time proportional to tokens. Third, the paradigm is prior work. Magnifying a question-relevant region at fixed image dimensions was published as AttWarp (Dalal et al., 2026). This attempt pastes the region, which AttWarp does not, and pasting and our box-shaped resampling fail in different places (Section 6.3). The later grids of Appendices D.6 and D.7 support the scientific claim that localization quality binds on natural scenes; they do not overturn this verdict, because the detector is still not free and the paradigm is prior work.

G.5 A test of the break-even framework that we do not pass

The two foveation attempts give two localizers of different quality on the same questions, which allows a cross-localizer test: calibrate u^+ and u^- on one localizer, then predict the other's gain from its hit rate alone.

Table 26: Testing the break-even framework as a predictor: calibrate the payoff on one localizer, then predict the other's gain from its hit rate alone. The verdict compares the predicted point with the measured interval and ignores the prediction's own uncertainty.

Model	Calibrated on → predicts	Predicted	Measured	Verdict
Qwen2.5-VL-7B	OWLv2 → attention	$+0.74$	$+1.57$ $[-5.24, +8.38]$	consistent
Qwen2.5-VL-7B	attention → OWLv2	$+11.38$	$+8.38$ $[+2.09, +15.18]$	consistent
Qwen2.5-VL-3B	OWLv2 → attention	$+5.34$	-4.19 $[-9.42, +1.05]$	**inconsistent**
Qwen2.5-VL-3B	attention → OWLv2	$+2.86$	$+8.38$ $[+2.62, +14.66]$	consistent

The framework survives on 7B and fails on 3B. There OWLv2-base's misses gain on average (a negative loss, $u^- = -3.45$), so the payoff of a miss calibrated on one localizer does not carry over to the other. With intervals 10 to 14 points wide at $n = 191$, agreement on 7B is a weak test. The break-even rate therefore remains a per-configuration diagnostic, not a validated predictor.

G.6 The pattern

Multi-layer fusion and gradient×attention fusion gained significantly on V*Bench ($+13.09^*$ and $+15.18^*$) and failed to transfer. Both selected their configuration with ground-truth boxes, which

exist only on V*Bench, so any method tuned there inherits that benchmark's idiosyncrasies. None of the remaining attempts passes the gate, and the one that raises the attention hit rate furthest, the four-scale union, reaches 69.6% only by spending 3,840 localization tokens against ViCrop's 490. Signals from the model's own forward pass reach the break-even hit rate only on the benchmark they were tuned on, or at a localization cost the break-even rate does not account for. The external detector we tried reaches it but does not survive a compute-matched, multi-benchmark gate. Two of our initial ideas, attention-layer selection and in-place magnification, turned out to be prior work (Zhang et al., 2025a; Dalal et al., 2026).

H The Attention Kernel as an Unreported Variable

While reconciling two runs of the same configuration we found a 2.09-point discrepancy in the no-crop baseline. The cause was the attention implementation. One script requested `eager` and the other took the library default (`sdpa`). Everything else (model, weights, data, prompt, visual-token budget) was identical, and the two runs consumed exactly the same number of visual tokens on every question.

Table 27: Two runs of Qwen2.5-VL-7B on V*Bench at 245 visual tokens that differ only in the attention implementation (`eager` or `sdpa`). Rows give accuracy and the number of the 191 questions on which the two runs disagree in the extracted answer, in the generated text and in the visual-token count.

	`eager`	`sdpa`
Accuracy (V*Bench, Qwen2.5-VL-7B, 245 tokens)	53.40	51.31
Per-question answer disagreement	16/191 = 8.4%	
Generated-text disagreement	27/191 = 14.1%	
Visual token disagreement	0/191	

The size of this effect is comparable to several differences reported as findings in this literature, and to several of the effects we report. We therefore hold the attention implementation fixed within every comparison and list in Appendix A.3 which experiments use which. We flag it for the same reason we flag the token budget. Like the budget, it is an implementation choice that moves benchmark numbers by as much as some of the effects being measured and is not usually reported.

I Extended Related Work

I.1 The reinforcement-learned zoom line

DeepEyes (Zheng et al., 2026) trains the model with reinforcement learning to call a zoom tool and reports large gains on the benchmarks we use. MED (Ma et al., 2026) decomposes the gains of crop-and-zoom RL into intrinsic capability improvement and tool-induced benefit, and concludes that intrinsic learning dominates while RL mainly reduces the harm the tool causes. MED does not match compute, and we do not separate intrinsic from tool-induced effects. We do not apply our protocol to this line, for the reason given in Appendix I.2.

I.2 The regime these results concern

The no-crop baseline improves with visual tokens only until the image runs out of detail. From the budget curves, that saturation point lies between 729 and 1,015 visual tokens on six of the seven configurations with full curves, and at 3,524 on Qwen2.5-VL-7B / V*Bench. At ViCrop's released operating point, a 256-token budget per image, the method's answering call uses 0.11 to 0.54 of the saturation budget, where the baseline still has headroom and an uncounted token advantage buys the most. The RL-trained line operates elsewhere. DeepEyes reports a no-tool baseline of 71.2 on V*Bench with Qwen2.5-VL-7B, and our budget sweep on that backbone, run with `eager` attention, tops out at 75.9% and stops improving beyond roughly 3,500 visual tokens. The compute DeepEyes's tool calls spend therefore cannot be returned to a baseline that has no more pixels to render. The 79.06% that the unresized native-resolution input reaches in Table 23 is not a point on this curve. It was measured with the library-default attention kernel on images that were not resized, so it cannot

be compared with the curve directly. Being past saturation does not by itself make cropping pay. Across seven configurations with full curves, the within-configuration slope of compute-matched gain against budget-over-saturation is positive in three and negative in four (sign test $p = 1.0$), and all six ViCrop cells above saturation are negative (-1.05, -1.75, -0.38, -0.58, -7.85, -10.75).

I.3 Dynamic resolution and per-image budgets

Qwen2.5-VL trains a native dynamic-resolution encoder and exposes a pixel cap that maps to a token count (Bai et al., 2025b); Qwen3-VL keeps that design at a different patch granularity (Bai et al., 2025a); InternVL3 splits the image into fixed 448×448 tiles, so attainable token counts are quantized (Zhu et al., 2025). In every case the cap applies per image, so a method that supplies two images consumes more visual tokens than a baseline given one at the same nominal setting, a difference the reported configuration does not show (Section 3.1).

I.4 What is borrowed

Compute-matched evaluation comes from the token-pruning literature. Attention-layer selection and the oracle crop are ViCrop's (Zhang et al., 2025a), the intersection-over-ground-truth measure and its case against IoU for crops follow Rodriguez & Rohrbach (2026), reallocating a fixed budget between global context and local detail is from Qian et al. (2025), and the observation that a correct crop is worth less at larger budgets is from Zhao et al. (2026). Non-uniform sampling of a fixed budget is AttWarp's (Dalal et al., 2026). Our equal-token experiments paste the magnified region over the image, which AttWarp does not do; the comparison between pasting and resampling (Section 6.3) is ours.