VISLENS: SINGLE-PASS INTERPRETABLE VISUAL SEARCH FOR MULTIMODAL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) struggle with fine-grained visual search, where a question concerns a small object in a high-resolution image. Existing remedies fall into two families. Training-free methods zoom in using attention or confidence scores but query the MLLM several times per example. Models trained with reinforcement learning (RL) to call cropping tools replace these queries with multi-turn rollouts but their tool calls are hard to control or interpret. To close this gap, we propose *VisLens* (Visual Focus via Logit Lens), which locates the target by reading the MLLM's own visual tokens through the logit lens, the projection of a hidden state through the LLM head onto the vocabulary. VisLens adds a lightweight tuned lens that maps early hidden states into the late hidden-state space, so visual tokens can be already read out from early layers. These tokens are matched to the target words in the query to produce a crop of the relevant region, which is fed back in alongside the original image to generate the final answer. The whole process, from decoding to answer, completes in a single forward pass with no repeated queries. Across three backbones, VisLens gains 8.6 to 12.1 points on V*Bench and HR-Bench and lies on the accuracy-latency Pareto frontier. It runs 6.8–$16.9\times$ faster than RL tool-use models and up to $28.5\times$ faster than training-free multi-pass search methods.

1 INTRODUCTION

Multimodal large language models (MLLMs) perform well on coarse tasks such as captioning and general visual question answering, but they struggle with fine-grained visual search, where the queried object fills only a tiny fraction of a high-resolution image (Wu & Xie, 2024; Wang et al., 2025; Zhang et al., 2025a). Under a fixed visual token budget, the target's signal is spread over a few tokens and easily lost among background content. Finding such small details reliably matters in applications such as document analysis, surveillance, and autonomous driving (Zhang et al., 2025c).

Existing remedies fall into two families, each with a clear cost. Training-free methods such as ViCrop (Zhang et al., 2025a), FOCUS (Zhong et al., 2026), ZoomEye (Shen et al., 2025), and UG-Search (Kim et al., 2025) read attention patterns or uncertainty scores to propose candidate regions, zoom in, and query the model again. Each example therefore costs several MLLM queries, which limits their use in time-sensitive settings. The second family trains the MLLM with reinforcement learning (RL) to call cropping or zooming tools, as in DeepEyes (Zheng et al., 2025), DeepEyesV2 (Hong et al., 2026), Thyme (Zhang et al., 2025b), and Mini-o3 (Lai et al., 2025). These RL tool-use models require costly training on task-specific tool-use data while their multi-turn rollouts still take several seconds per example during inference. Moreover, their tool calls are hard to interpret, since they rarely explain why a region was chosen.

Our key insight is that locating the target needs neither repeated queries nor a learned tool policy, because the MLLM's own visual tokens already encode what sits at each image location. This holds even for the tokens entering the LLM, before any LLM layer has processed them. Decoding these tokens into words and matching the words against the target named in the query recovers the target's location directly.

We operationalize this insight in *VisLens* (Visual Focus via Logit Lens), a method built on the logit lens (nostalgebraist, 2020; Wang et al., 2022; Wendler et al., 2024; Wang, 2025), a technique widely used to interpret the semantic space of LLMs and recently shown to work similarly well for visual

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.

tokens in MLLMs (Phukan et al., 2025; Jiang et al., 2025; Neo et al., 2025). The logit lens projects a hidden state at any layer through the LLM's output head to approximate what the model would predict at that point. VisLens uses this projection to decode the semantics of each image region, match the decoded tokens against the target words from the query, and merge the best-matching regions into crop boxes through clustering. These boxes are then fed alongside the original image to produce the final answer. Because the crop comes from an early MLLM layer, the whole process still fits in a single forward pass, avoiding the latency of repeated queries while staying transparent, as each localization decision traces back to specific decoded tokens.

However, applying the logit lens directly to early layers is unreliable as the output head is optimized for final-layer representations. VisLens therefore trains a lightweight tuned lens (Belrose et al., 2023), a small MLP that maps early hidden states into the final layer's representational space before decoding. This mapping is cheap to train, needs no labels, and leaves the underlying MLLM frozen. Together these properties place VisLens in the corner of Table 1 that neither prior family reaches. As a result, VisLens consistently improves accuracy at a fraction of the latency of previous baselines. Multi-pass search matches it only with an order of magnitude more time, and RL methods remain several times slower.

Table 1: **Comparison of visual search paradigms.** VisLens answers in a single pass, avoids costly task-specific training, and yields interpretable localization.

Method	Training-Free	RL Tool-Use	VisLens
Single Forward Pass	✗	✗	✓
No Costly Training	✓	✗	✓
Results Interpretable	✓	✗	✓

In summary, our contributions are as follows: (1) We propose VisLens, a single-pass and interpretable visual search method built on the logit lens. (2) VisLens lies on the accuracy-latency Pareto frontier against both training-free search and RL tool-use models. (3) We show that the decoded tokens explain both successes and failures of VisLens, and that its failures follow how the backbone itself encodes the target.

2 Related Work

Visual search in MLLMs. V*Bench formalized visual search as answering questions about small targets in high-resolution images, and its SEAL framework pairs the MLLM with a separately trained search model (Wu & Xie, 2024). Training-free methods derive candidate regions from the MLLM itself and query it again on each. ViCrop (Zhang et al., 2025a) crops from attention maps, ZoomEye (Shen et al., 2025) searches a tree of nested regions scored by the model, and UG-Search (Kim et al., 2025) ranks regions by the uncertainty of the model's output. FOCUS (Zhong et al., 2026) extracts target objects from the question and localizes them with a relevance map computed from the KV cache. RL tool-use models instead learn where to crop. DeepEyes (Zheng et al., 2025) interleaves reasoning with zoom-in calls, DeepEyesV2 (Hong et al., 2026) broadens the tool set, Thyme (Zhang et al., 2025b) executes image-processing code, and Mini-o3 (Lai et al., 2025) scales the number of interaction turns. VisLens differs from both families in that it localizes the target from decoded words before any LLM layer runs.

Logit lens in MLLMs. The logit lens reads an intermediate prediction of an LLM by projecting its hidden state through the output head (nostalgebraist, 2020). It has been used to study circuits (Wang et al., 2022), multilingual representations (Wendler et al., 2024), and modern architectures (Wang, 2025). Since the output head is fit to final-layer states, the raw lens is unreliable at early layers, which the tuned lens (Belrose et al., 2023) corrects with a learned affine map per layer. In MLLMs, decoded visual tokens name objects at their true positions (Neo et al., 2025), and the lens has been used to detect hallucinations (Phukan et al., 2025; Jiang et al., 2025) as well as to localize objects and edit out hallucinated content (Jiang et al., 2025). VisLens makes the tuned lens the search mechanism itself so that search and answering share one forward pass.

3 Visual Focus via Logit Lens (VisLens)

Task definition. Visual search maps a high-resolution image I and a query q about one or more target objects to an answer y. Under a fixed token budget, a small target has its signal spread thin

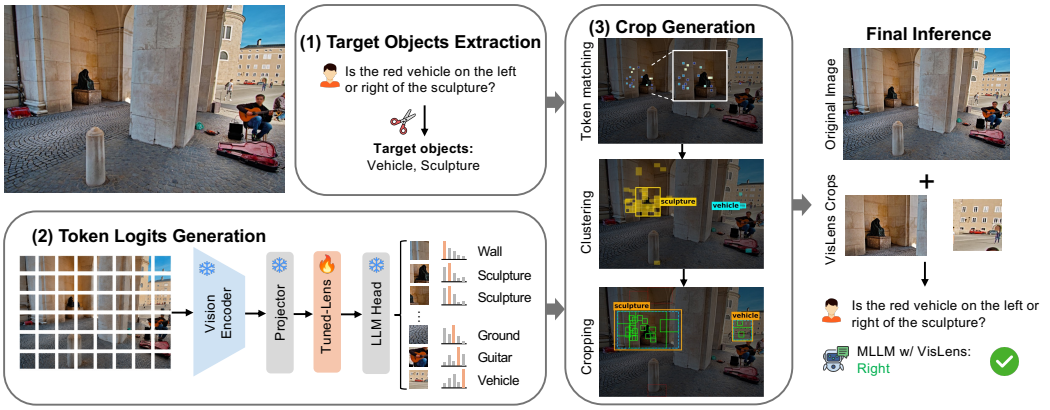


Figure 1: **VisLens pipeline.** (1) Target objects extraction pulls the localizable target objects from the query. (2) Token logit generation decodes the pre-LLM visual states through the tuned lens and LLM head into top-K tokens per cell. (3) Crop generation turns this map into one crop per target by matching, clustering, and merging. A final inference step then feeds the crops and the original image to the same frozen MLLM for the answer. Only the tuned lens is trained.

across the visual tokens, whereas a focused crop restores the missing detail. VisLens therefore recovers a set of crops $\mathcal{I}_c$ from I and q alone and answers with $y = \mathrm{MLLM}(I, \mathcal{I}_c, q)$.

Logit lens preliminaries. The MLLM encodes I into N visual tokens, each tied to one cell of the image grid. Let $h_\ell^{(i)} \in \mathbb{R}^d$ be the hidden state of visual token i after LLM layer ℓ, where $h_0^{(i)}$ is the projector output entering the first layer and $h_L^{(i)}$ is the final-layer state. The logit lens applies the frozen LLM head, the final normalization followed by the unembedding W_U, to such a state,

$$\phi(h) = \mathrm{softmax}\big(W_U\ \mathrm{LN}(h)\big) \in \Delta^{|\mathcal{V}|},$$

approximating the prediction the model would make if it decoded from that layer. At a visual position, the top entries of $\phi\big(h_\ell^{(i)}\big)$ name the object or attribute encoded there (Phukan et al., 2025; Jiang et al., 2025; Neo et al., 2025), so the N distributions form a spatial map of where each concept is represented.

Tuned lens. Logit-lens readouts degrade at early layers because W_U is trained to decode only h_L, while later layers progressively change the basis and scale of the residual stream. We therefore insert a lightweight translator into the final-layer space, $g_\ell(h) = h + \mathrm{MLP}_\ell(h)$, a residual bottleneck MLP initialized near identity ($\mathrm{MLP}_\ell \approx 0$) so that training starts from the raw logit lens. This gives the tuned-lens readout $\mathrm{TL}_\ell(h) = \phi\big(g_\ell(h)\big)$. We train one g_ℓ per source layer by distillation in vocabulary space, restricted to the visual positions $\mathcal{P}_{\mathrm{vis}}$:

$$\min_{\theta_\ell}\ \mathbb{E}_{(I,q)}\,\mathbb{E}_{i\in\mathcal{P}_{\mathrm{vis}}}\ D_{\mathrm{KL}}\Big(\phi\big(h_L^{(i)}\big)\ \big\|\ \phi\big(g_\ell(h_\ell^{(i)})\big)\Big).$$

The teacher is the model's own final-layer distribution, so no labels are required, and the objective directly optimizes the distribution used for decoding rather than regressing hidden states. The MLLM stays frozen and only θ_ℓ is updated (about 0.05% of the backbone parameters), so the translator learns the change of basis that lets the frozen head read early visual states without any task supervision. Our design follows the tuned lens of Belrose et al. (2023) but differs in two ways: the translator is a nonlinear bottleneck MLP rather than an affine map, and it is fit on visual positions alone. We use $\ell = 0$ by default, which reads the visual tokens before any LLM layer runs and localizes as well as deeper source layers (Table 8).

VisLens pipeline. VisLens turns the image and query into crops through three stages before the final inference (Figure 1).

(1) Target object extraction. We part-of-speech tag q and use WordNet (Miller, 1995) to keep the content nouns that denote localizable objects, dropping attribute and observer words such as *color* and *camera*. Adjacent nouns merge into one phrase such as *fire hydrant*, so each phrase names a single target. This gives $P = \{p_1, \ldots, p_M\}$, such as $\{\text{vehicle}, \text{sculpture}\}$ in Figure 1.

(2) Token logit generation. The vision encoder and projector turn I into the states $h_0^{(i)}$. The tuned lens maps each state into the final-layer space, and the LLM head decodes every cell into its top-K (token, probability) pairs $S_i = \text{top-}K\big(\text{TL}_\ell(h_\ell^{(i)})\big)$. With $\ell = 0$, no LLM layer runs in this stage.

(3) Crop generation. This stage turns $\{S_i\}_{i=1}^N$ into one box per phrase by matching, clustering, and merging.

Token matching scores each cell against each phrase p_j with a three-tier cascade, giving a score map $H_j \in \mathbb{R}^N$. A tier runs only if all earlier tiers leave every cell unmatched. The *exact* tier matches a cell when one of its top-K tokens t equals a word of p_j after case and punctuation normalization, and assigns it the lens probability π of that token, $H_j[i] = \max_{(t,\pi)\in S_i} \pi\, \mathbb{1}[t \in p_j]$. The *synonym* tier expands p_j with the WordNet lemmas, hypernyms, and hyponyms of its dominant noun sense (*handbag→bag*, *purse*) and repeats the exact test. The *soft* tier covers the case where no surface form matches, most likely because the target word fell below the top-K cutoff rather than being absent from the state. It skips decoding and scores each cell in embedding space against the unembedding rows of $\mathcal{T}_j$, the sub-word ids of the words in p_j, as $H_j[i] = \max_{t\in\mathcal{T}_j} \cos\big(g_\ell(h_\ell^{(i)}) - \bar{u},\, W_U[t,:] - \bar{u}\big)$. Here $\bar{u}$ is the mean unembedding row, whose subtraction removes a shared direction that otherwise compresses all cosines into a narrow band. Since this score is dense, we keep only the top-scoring cells and shift their scores so that the weakest retained one becomes zero, restoring the sparsity that clustering assumes.

Clustering treats H_j the same way whichever tier produced it. It groups the matched cells $\{i : H_j[i] > 0\}$ into connected components under 8-adjacency (Rosenfeld & Pfaltz, 1966), which links cells at most one cell apart, and scores each component C by its mass $\sum_{i\in C} H_j[i]$.

Merging adds the components of p_j to its box in decreasing order of score and stops before the enclosing box would exceed a fraction τ of the image area. Padding this box by one cell gives B_j, and VisLens crops each box independently, $\mathcal{I}_c = \{\text{Crop}(I, B_j)\}_{j=1}^M$. For backbones that encode both a low-resolution thumbnail and high-resolution tiles, namely LLaVA-OneVision (Li et al., 2024) and InternVL3 (Zhu et al., 2025), we take the union of the boxes from the two grids in pixel space before cropping.

Final inference. The frozen MLLM answers from the crops together with the original image, which keeps the spatial relations between targets readable. If no crop is produced, it answers from I alone. Because stages (1) to (3) run before the first LLM layer, the LLM processes the query only once, and the crops add only a vision encoder call.

4 Experiments

4.1 Setup

Models. We apply VisLens to three open-source MLLMs: LLaVA-OneVision (Li et al., 2024), Qwen2.5-VL (Bai et al., 2025), and InternVL3 (Zhu et al., 2025). For each backbone, we train the tuned lens once at the default source layer $\ell = 0$ and use it unchanged on every benchmark.

Benchmarks. We evaluate on visual search benchmarks whose targets fill a small fraction of a high-resolution scene: V*Bench (Wu & Xie, 2024), HR-Bench 4K and 8K (Wang et al., 2025), MME-RealWorld-Lite (Zhang et al., 2025c), and VisualProbe (Lai et al., 2025). To check that VisLens does not hurt questions about normal-size objects, we also test on the standard VQA benchmarks A-OKVQA (Schwenk et al., 2022), MMBench (Liu et al., 2024b), AMBER (Wang et al., 2023), and RealWorldQA (Liu & Zhang, 2025). Finally, we apply VisLens to VNBench (Zhao et al., 2024), where the task is to find key frames in a long video.

Baselines. We first compare VisLens with its unmodified backbone (Section 4.2). We then compare it on the accuracy-latency trade-off with training-free search on LLaVA-OV-7B, namely ViCrop (Zhang et al., 2025a), ZoomEye (Shen et al., 2025), and UG-Search (Kim et al., 2025) (Figure 2), and with RL tool-use models built on Qwen2.5-VL-7B, namely Thyme (Zhang et al., 2025b), Mini-o3 (Lai et al., 2025), DeepEyes (Zheng et al., 2025), and DeepEyesV2 (Hong et al., 2026) (Figure 3). Each comparison shares one backbone, so differences reflect the search mechanism rather than the underlying MLLM.

Table 2: **Visual search results.** Accuracy (%) of each backbone with and without VisLens. Subscripts in green denote the gain over the backbone.

Model	V*Bench	HR-Bench 4K	HR-Bench 8K	MME-RW-Lite	VisualProbe
LLaVA-OV-7B	73.3	63.8	58.5	47.2	28.3
w/ **VisLens**	83.2$_{9.9}$	72.4$_{8.6}$	69.6$_{11.1}$	49.6$_{2.4}$	35.7$_{7.4}$
Qwen2.5-VL-7B	64.9	62.1	56.5	40.4	25.8
w/ **VisLens**	76.4$_{11.5}$	73.2$_{11.1}$	68.6$_{12.1}$	43.5$_{3.1}$	37.7$_{11.9}$
InternVL3-8B	71.7	69.9	61.5	47.5	26.4
w/ **VisLens**	83.8$_{12.1}$	79.4$_{9.5}$	71.0$_{9.5}$	49.8$_{2.3}$	38.8$_{12.4}$

Metrics. We report accuracy and the average end-to-end wall-clock latency per question. Since every method trades extra computation for better localization, latency shows the cost at which each method reaches its accuracy.

4.2 Main Results

Gains hold across backbones. VisLens improves all three backbones on every benchmark in Table 2. On V*Bench and HR-Bench, the gain ranges from 8.6 to 12.1 points, even though the backbones differ in encoder, tiling scheme, and base accuracy. The gain is therefore not tied to one architecture.

Gains grow as targets shrink. Moving from HR-Bench 4K to 8K makes each target smaller relative to the encoded image, and every backbone loses 5 to 8 points. The VisLens gain does not shrink with it. It grows for LLaVA-OV-7B and Qwen2.5-VL-7B and stays level for InternVL3-8B. VisualProbe, with small targets among many similar distractors, gives the largest relative gains, close to half the backbone accuracy for Qwen2.5-VL-7B and InternVL3-8B. The harder a target is to resolve in the full image, the more a focused crop helps.

Where the gains on MME-RealWorld-Lite come from. Unlike our other benchmarks, MME-RealWorld-Lite is not built exclusively around small objects, and Table 6 shows that its overall gain of 2.3 to 3.1 points averages over very different domains. The gains concentrate where the question names a concrete object in a cluttered scene. Autonomous driving improves on every backbone, by 2.8 to 5.7 points on perception and 1.5 to 5.0 on reasoning, and OCR perception improves by 0.8 to 4.4. The gains disappear in two places. Remote sensing loses up to 4.0 points on two of three backbones, likely because the query noun matches many near-identical cells in aerial views and no single crop isolates the target. Diagram and table reasoning is similarly mixed (-2.0 to $+3.0$), since the answer depends on the whole layout rather than one region.

4.3 Accuracy-Latency Trade-off

Against training-free search. Figure 2 shows how multi-pass search gains accuracy with more time. In the low-latency region only VisLens and ViCrop remain, and at similar latency VisLens is more accurate on every benchmark. Both derive a crop from the model's internal signals without search, so this gap suggests that decoded tokens locate the target better than attention maps. ZoomEye and UG-Search start below VisLens at their cheapest settings, and ZoomEye's cheapest setting falls below even ViCrop. Multi-pass search matches VisLens only at about $11\times$ its latency on V*Bench and $28.5\times$ on HR-Bench 8K, and no setting we evaluate reaches it on HR-Bench 4K. Although a large search can still buy more accuracy as ZoomEye shows on V*Bench, VisLens holds the low-latency end of the Pareto frontier, which no multi-pass setting reaches.

Against RL tool-use models. Figure 3 compares VisLens with models that learn a tool policy on the same Qwen2.5-VL-7B backbone. Each of them runs a multi-turn rollout and needs several seconds per example, whereas VisLens answers in about half a second, 6.8–$16.9\times$ faster. In accuracy, VisLens beats Thyme on all three benchmarks. It trails Mini-o3 on V*Bench, is level with it on HR-Bench 4K, and passes it on HR-Bench 8K. DeepEyes and DeepEyesV2 stay ahead, but their lead is largest on V*Bench and narrows on HR-Bench. We also find that rollout cost does not track accu-

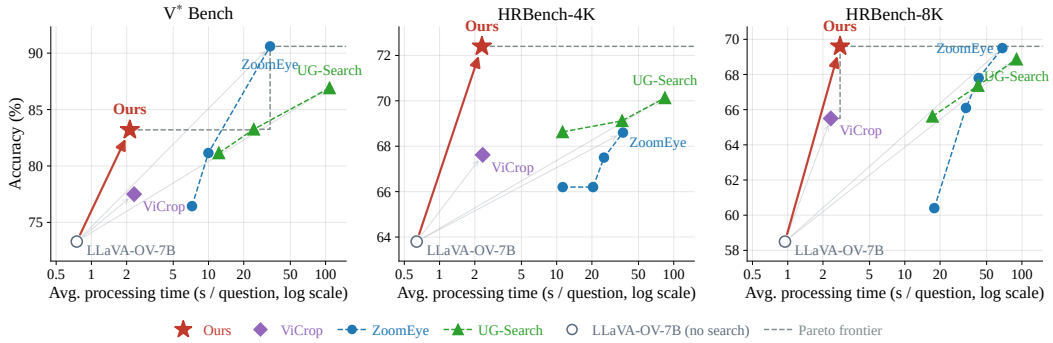


Figure 2: **Accuracy and latency against training-free search** (LLaVA-OV-7B). Lines from the backbone without search (open circle) show the accuracy each method gains for the latency it adds. Dashed curves trace ZoomEye and UG-Search at growing search budgets, and the gray dashed line marks the Pareto frontier. VisLens lies on the frontier in every panel. The x-axis is logarithmic.

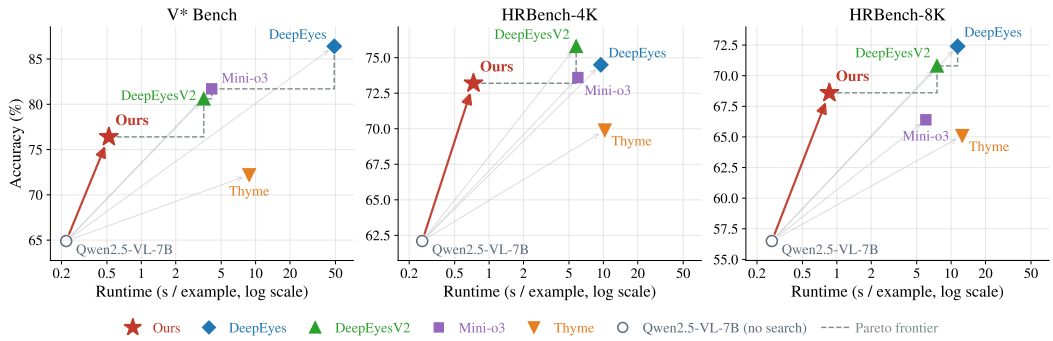


Figure 3: **Accuracy and latency against RL tool-use models** (Qwen2.5-VL-7B). Lines from the backbone without search (open circle) show the accuracy each method gains for the latency it adds. VisLens has the steepest line and lies on the Pareto frontier (gray dashed) in every panel, gaining 8–10$\times$ more accuracy per added second than any baseline. The x-axis is logarithmic.

racy. Thyme executes its generated code in a sandbox and is among the slowest models, yet it is the least accurate RL model in every panel. Finally, all four models need RL training on purpose-built tool-use data, which VisLens does not.

4.4 Robustness on Standard VQA and Transfer to Video

No degradation on standard VQA. The left block of Table 3 covers benchmarks whose questions concern relatively large objects, where perception is not the bottleneck. VisLens changes accuracy by less than one point on every benchmark and backbone, and the only drop is 0.9 points for InternVL3-8B on A-OKVQA. VisLens therefore preserves general VQA performance. On AMBER, which measures hallucination, accuracy stays level or rises slightly on all three backbones, so the added crops do not lead the model to report absent objects.

Search transfers to video. VNBench (Zhao et al., 2024) inserts synthetic needles into long videos and asks about their identity, order, and count. Each needle appears in only one or a few frames and covers a small part of them, while the remaining frames act as distractors. Answering therefore depends on finding the right frames rather than on handling a long context, which makes it visual search along time instead of space. VisLens applies token logit generation and token matching (Section 3) to every sampled frame, scores each frame by its matched tokens, and gives the highest-scoring frames to the model for the final answer. No video-specific training is involved. VisLens gains 4.3 to 11.0 points under the strict protocol of the benchmark, which counts a question as correct only if all four option rotations are answered correctly. The same pipeline thus works as a general search procedure over visual input rather than an image-specific heuristic.

Table 3: **Standard VQA and video search.** Accuracy (%) with and without VisLens. VisLens is never trained on video. Green and red subscripts mark gains and drops over the backbone.

	Standard VQA				Video
Model	**A-OKVQA**	**MMBench**	**AMBER**	**RealWorldQA**	**VNBench**
LLaVA-OV-7B	90.8	85.8	87.8	67.1	53.1
w/ VisLens	90.9$_{+0.1}$	85.8$_{+0.0}$	88.6$_{+0.8}$	67.3$_{+0.2}$	64.1$_{+11.0}$
Qwen2.5-VL-7B	88.2	87.9	87.4	68.8	63.3
w/ VisLens	88.6$_{+0.4}$	88.2$_{+0.3}$	87.5$_{+0.1}$	69.2$_{+0.4}$	74.0$_{+10.7}$
InternVL3-8B	89.1	87.9	87.8	68.3	64.6
w/ VisLens	88.2$_{-0.9}$	88.3$_{+0.4}$	88.2$_{+0.4}$	68.5$_{+0.2}$	68.9$_{+4.3}$

Table 4: **VisLens versus external detectors** on V*Bench. Each detector replaces token logit generation and crop generation, takes each extracted target as a text prompt, and runs at its default threshold (e.g., 0.5 for SAM3). Green and red subscripts mark gains and drops over the backbone.

Method	**LLaVA-OV-7B**	**Qwen2.5-VL-7B**	**InternVL3-8B**
Baseline	73.3	64.9	71.7
Grounding-DINO	72.3$_{-1.0}$	67.5$_{+2.6}$	71.7$_{+0.0}$
SAM3	74.3$_{+1.0}$	71.2$_{+6.3}$	77.0$_{+5.3}$
VisLens	83.2$_{+9.9}$	76.4$_{+11.5}$	83.8$_{+12.1}$

4.5 VisLens versus External Detectors

Setup. To test whether the gain comes from cropping alone, we replace token logit generation and crop generation with an open-vocabulary detector and keep target object extraction and final inference unchanged. Each extracted target is passed to the detector as a text prompt, and its highest-confidence box above the default threshold becomes the crop. We use Grounding-DINO (Liu et al., 2024a) and the stronger SAM3 (Carion et al., 2025), at their default confidence of 0.5.

Results. Table 4 shows that detector quality matters but does not close the gap. Grounding-DINO changes accuracy by -1.0 to $+2.6$ points and SAM3 by $+1.0$ to $+6.3$, while VisLens gains $+9.9$ to $+12.1$. Much of the gap to VisLens comes from recall. Both detectors return no box on roughly 40% of queries (39.7% for Grounding-DINO and 41.0% for SAM3), whereas VisLens produces a crop for more than 90%. Since the two detectors miss equally often, the advantage of SAM3 over Grounding-DINO comes from the quality of the boxes it does return.

Implication. This suggests that the gain comes from the crop generation pipeline of VisLens via logit lens rather than from cropping itself. VisLens reads object evidence already present in the MLLM's own visual tokens instead of asking a separate model to recognize the target. The Grounding-DINO result on LLaVA-OV-7B also shows that a crop is not automatically useful, since an imprecise region can leave the backbone worse off than no search at all.

5 Interpretability Analysis

5.1 When the Target Is Decoded but Underused

Figure 4 shows two questions that the backbone answers wrongly without search. In both, the tuned-lens readout at the target region already names the target. The information is present in the visual tokens, but the backbone does not use it when it sees only the full image.

Single-object attribute. The left question asks for the color of the pants worn by a person, and the backbone answers that no person is in the image. The tuned-lens tokens at a small region of the field include `pants`, `shorts`, `trousers`, and `black`, so the readout contains both the target and the answer. VisLens crops this region and gives it to the model with the original image, and the model answers black.

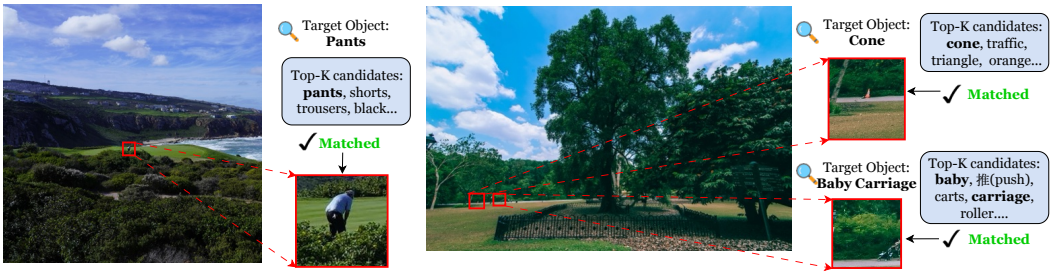


Figure 4: **Successful cases.** For each extracted target, the tuned-lens top-K tokens at the matched region and the resulting crop (red). The backbone answers both questions wrongly, and VisLens corrects them.

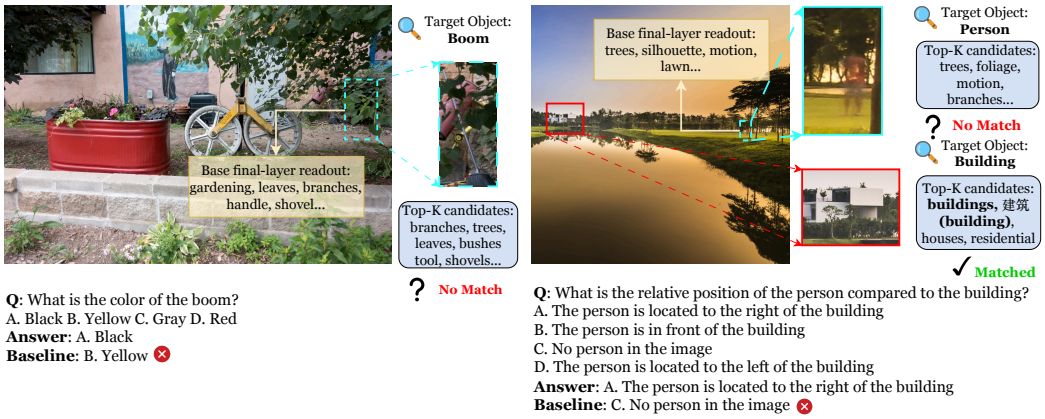


Figure 5: **Failure cases.** For each target, the tuned-lens top-K tokens and the base model's final-layer readout at the target region. Neither readout contains the target word for the boom (left) or the person (right), so VisLens cannot localize them, while the building (right) is matched.

Multiple targets. The right question asks whether a baby carriage is on the left or right of a cone. VisLens matches both targets. The cone region decodes `cone`, `traffic`, `triangle`, and `orange`, and the carriage region decodes `baby`, `carts`, `carriage`, and a Chinese token meaning *push*. Each target receives its own crop, and the original image stays in the prompt so that the model can still read their spatial relation. The backbone answers left, and VisLens corrects it to right.

In both cases the backbone already encodes the target, and VisLens turns this readable evidence into crops the model can use.

5.2 When the Target Is Not Decoded

The tokens that drive localization also explain its failures. When VisLens misses a target, the readout shows what the model encodes in its place. Figure 5 shows two such cases, each with the base model's final-layer readout at the same region.

Target encoded as related concepts. The left question asks for the color of a boom. The tuned-lens tokens at the boom include `branches`, `trees`, `leaves`, `bushes`, `tool`, and `shovels`. The final-layer readout gives `gardening`, `leaves`, `branches`, `handle`, and `shovel`. Both describe the region as garden plants and tools, and neither contains a word that matches *boom*, so VisLens cannot localize it. The backbone answers yellow instead of black.

Target encoded as background. The right question asks for the position of a person relative to a building. The building region decodes `buildings`, `houses`, and `residential` and is

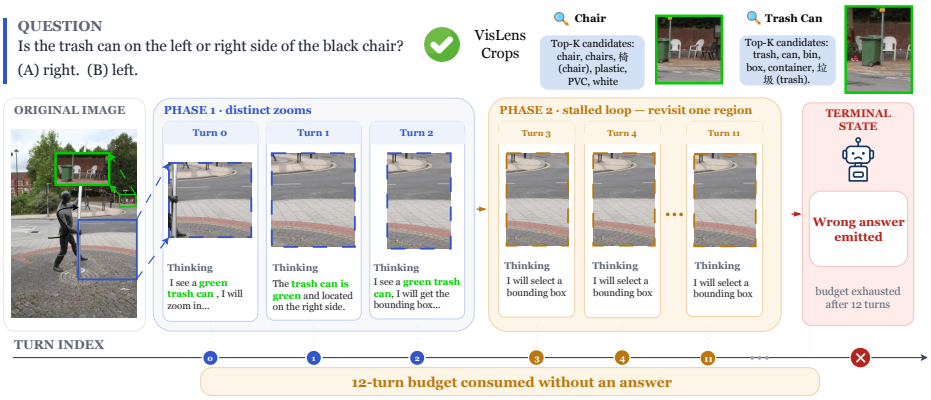


Figure 6: **Tool-use model failure.** Mini-o3 names the green trash can at turn 0, but every crop covers the pavement in the lower right (blue) rather than the target region (green). From turn 3 it revisits the same region until the 12-turn budget runs out and a wrong answer is returned. VisLens (top right) crops both targets in a single pass and answers correctly.

matched. The person, a small figure against the sunset, decodes as `trees`, `foliage`, `motion`, and `branches`, and the final-layer readout gives `trees`, `silhouette`, `motion`, and `lawn`, where only `silhouette` hints at a figure. VisLens crops only the building, so the relation cannot be resolved. The backbone answers that no person is in the image.

Failures follow the backbone. In both cases the tuned-lens readout agrees with the final-layer readout, and the backbone also fails without search. The miss is therefore not added by the lens but reflects how the backbone itself encodes the region. Table 8 shows the same beyond these cases, since the tuned lens detects small COCO objects nearly as often as the final-layer readout (0.567 against 0.582). The readout also tells the two failures apart, with related names in the first and background in the second.

5.3 Where RL Tool Use Fails and VisLens Succeeds

RL tool-use models provide no comparable explanation. Figure 6 shows Mini-o3 (Lai et al., 2025) asked whether a trash can is on the left or right of a black chair. At turn 0 its reasoning already names the target, a green trash can, yet its first three crops zoom into different parts of the pavement in the lower right, away from the trash can. From turn 3 on, the model repeats the intention to select a bounding box without moving the crop, until the 12-turn budget runs out and it returns a wrong answer. The trace shows that the reasoning and the crops disagree, but not why. On the same question, VisLens crops the two target successfully by decoding `trash`, `can`, `bin`, `container`, and their Chinese synonyms for the trash can, and `chair`, `chairs`, and `plastic` for the chair.

6 Conclusion and Limitations

We introduced VisLens, a single-pass and interpretable visual search method that decodes the visual tokens of a frozen MLLM through a lightweight tuned lens and crops the regions whose tokens match the query targets. Across three backbones, VisLens improves every visual search benchmark and lies on the accuracy-latency Pareto frontier against both training-free search and RL tool-use models, while preserving standard VQA accuracy and transferring to video. Because each crop comes from decoded tokens, its localization decisions and failures can be interpreted directly.

As VisLens relies on local semantics, it works best when the target is an object whose name can be read from nearby cells. It is less reliable when the evidence is textual, symbolic, or spread across the layout. In a high-resolution diagram, each cell holds only a fragment such as a digit or part of a line. The tuned lens then decodes these fragments rather than the queried concept, and wrong cell matches the target (Figure 8). Extending VisLens with structural grouping and relation-aware matching across cells is a direction for future work.

AI USE STATEMENT

We used Claude (Anthropic) and GPT (OpenAI) solely to refine the wording and grammar of author-written text. We did not use generative AI tools for any tasks with required disclosure (e.g., research ideation, methodology, implementation, data processing, analysis, or interpretation of results). All AI-assisted edits were reviewed by the authors, who take full responsibility for the content of this paper.

REFERENCES

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

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, and Guohai Xu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026.

Nick Jiang, Anish Kachinthaya, Suzanne Petryk, and Yossi Gandelsman. Interpreting and editing vision-language representations to mitigate hallucinations. In *International Conference on Learning Representations*, volume 2025, pp. 63582–63605, 2025.

Sanghwan Kim, Rui Xiao, Stephan Alaniz, Yongqin Xian, and Zeynep Akata. Training-free uncertainty guidance for complex visual tasks with mllms. *arXiv preprint arXiv:2510.00705*, 2025.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *arXiv preprint arXiv:2509.07969*, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ming Liu and Wensheng Zhang. Reasoning multimodal large language model: Data contamination and dynamic evaluation. *arXiv preprint arXiv:2506.07202*, 2025.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024a.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024b.

George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *International Conference on Learning Representations*, volume 2025, pp. 57172–57189, 2025.

nostalgebraist. interpreting GPT: the logit lens. LessWrong, August 2020. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Anirudh Phukan, Divyansh Divyansh, Harshit Kumar Morj, Vaishnavi Vaishnavi, Apoorv Saxena, and Koustava Goswami. Beyond logit lens: Contextual embeddings for robust hallucination detection & grounding in vlms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9661–9675, 2025.

Azriel Rosenfeld and John L Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4):471–494, 1966.

Gabriel Sarch, Linrong Cai, Qunzhong Wang, Haoyang Wu, Danqi Chen, and Zhuang Liu. Vero: An open rl recipe for general visual reasoning. *arXiv preprint arXiv:2604.04917*, 2026.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

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 *EMNLP*, 2025.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

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 *AAAI*, 2025.

Zhenyu Wang. Logitlens4llms: Extending logit lens analysis to modern large language models. *arXiv preprint arXiv:2503.11667*, 2025.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, 2024.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, 2024.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Mllms know where to look: Training-free perception of small visual details with multimodal llms. *ICLR*, 2025a.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025b.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *ICLR*, 2025c.

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic evaluator for video mllms. *arXiv preprint arXiv:2406.09367*, 2024.

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. *arXiv preprint arXiv:2505.14362*, 2025.

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. *Advances in Neural Information Processing Systems*, 38: 96498–96535, 2026.

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

A Details of Baselines

A.1 General MLLM Baselines

We evaluate VisLens on three open-source MLLMs that differ in visual encoder and resolution handling, so that the reported gains cannot be attributed to one design.

LLaVA-OneVision-7B (Li et al., 2024) pairs a SigLIP encoder with Qwen2-7B and uses AnyRes tiling. A high-resolution image is split into tiles that are encoded independently and concatenated with a downsampled view of the whole image, and each 384×384 view yields a 27×27 token grid. VisLens therefore sees a thumbnail grid and several tile grids for this backbone.

Qwen2.5-VL-7B (Bai et al., 2025) uses a native-resolution ViT with window attention and 2D-RoPE. Each visual token merges 2×2 patches of 14 pixels, so it covers a 28×28 pixel cell of the resized input, and a single grid covers the whole image with no separate thumbnail.

InternVL3-8B (Zhu et al., 2025) pairs InternViT-300M with Qwen2.5-7B and uses dynamic tiling into 448×448 tiles plus a thumbnail. Pixel shuffle reduces each tile to a 16×16 token grid, so like LLaVA-OneVision it emits both tile and thumbnail tokens.

A.2 Training-free methods

These methods need no additional training but query the MLLM more than once per question, which is the cost VisLens removes. All are run on LLaVA-OV-7B (Figure 2).

ViCrop (Zhang et al., 2025a) derives a crop from the MLLM's attention between the question and the image, or from its gradients, and answers again with the crop added to the input. It needs one pass to build the map and another to answer. Its cost is the closest to VisLens, which makes it the most direct training-free comparison.

ZoomEye (Shen et al., 2025) treats the image as a tree of nested regions, scores candidate regions with the MLLM itself, and descends into the most promising branch. Its cost grows with the search budget. We sweep its search depth and branching to obtain the operating points.

UG-Search (Kim et al., 2025) scores candidate crops by the entropy of the model's response and keeps the least uncertain one, based on the observation that uncertainty drops when the relevant evidence is visible.

A.3 RL Tool-Use Models

These methods train the MLLM with RL to invoke visual tools. They avoid repeated external queries but replace them with multi-turn rollouts. All four build on Qwen2.5-VL-7B (Figure 3).

DeepEyes (Zheng et al., 2025) is trained end-to-end with RL to interleave reasoning with zoom-in tool calls. At each turn it either answers or predicts a region to crop, and the crop is appended to its context for the next turn.

DeepEyesV2 (Hong et al., 2026) extends DeepEyes with code execution and web search in one reasoning loop. It is trained with a cold-start stage followed by RL, and it runs faster than DeepEyes in our measurements (Figure 3).

Thyme (Zhang et al., 2025b) generates executable code for image manipulation and computation, runs it in a sandbox, and reasons over the returned result. Every tool call therefore adds a sandbox round trip.

Mini-o3 (Lai et al., 2025) is trained for long-horizon visual search with at most six interaction turns and scales to tens of turns at test time. We cap it at 12 turns following the default, and it can exhaust this budget without converging (Figure 6).

B Details of Benchmarks

B.1 Visual search QA

V*Bench (Wu & Xie, 2024) contains 191 questions over high-resolution natural images in which the queried object occupies a small fraction of the scene, split into *attribute* questions about a single target and *spatial* questions about the relation between two targets.

HR-Bench (Wang et al., 2025) provides 800 questions each at 4K and 8K resolution, split into fine single-object perception (FSP) and fine cross-object perception (FCP). The 8K split is the hardest setting we evaluate: the target is smaller relative to the encoded resolution, which is where multi-pass search is most expensive and where VisLens gains most.

MME-RealWorld-Lite (Zhang et al., 2025c) contains 1,919 questions over 1,543 real-world images of about $2{,}800 \times 1{,}600$ pixels on average. They come from five domains, namely OCR in the wild, remote sensing, diagram and table, monitoring, and autonomous driving, and are divided into perception and reasoning. Unlike the two benchmarks above, it is not exclusively a small-target benchmark, so only part of it exercises the localization bottleneck.

VisualProbe (Lai et al., 2025) is the visual search benchmark released with Mini-o3 and built to require many interaction turns. Its targets are small, visually similar distractors are abundant, and its images are cluttered. Its 515 test questions are stratified into 141 easy, 268 medium, and 106 hard questions. We follow the Mini-o3 protocol with Qwen3-32B as the judge.

B.2 Standard VQA

To verify that VisLens does not degrade behavior on images without a small-target bottleneck, we evaluate on four standard VQA benchmarks.

A-OKVQA (Schwenk et al., 2022) requires commonsense and world knowledge rather than fine-grained perception, so it tests whether added crops distract the model.

MMBench (Liu et al., 2024b) covers a broad range of multimodal abilities with multiple-choice questions scored under CircularEval, which rotates the options.

AMBER (Wang et al., 2023) probes hallucination by asking whether named objects, attributes, and relations are present. It is the benchmark most sensitive to a spurious crop, since a wrongly focused region could lead the model to report absent content.

RealWorldQA (Liu & Zhang, 2025), released by xAI, contains 765 real-world images, many taken from vehicles, each with a question on basic spatial understanding.

B.3 Video Needle in a Haystack

VNBench (Zhao et al., 2024) inserts synthetic needles, such as subtitles or images, into videos of 10 to 180 seconds and asks about their identity (retrieval), order (ordering), or number (counting). Each task has three subtasks that differ in needle type and difficulty, giving 9 question types and 1,350 questions in total. All questions are four-way multiple choice. Following the benchmark protocol, a question counts as correct only if all four option-rotated presentations are answered correctly, which makes the metric strict and insensitive to option-position bias.

C Implementation Details of VisLens

Tuned lens. All three language models have $L = 28$ layers and hidden size $d = 3584$, so the lenses share one architecture and differ only in their weights. For each backbone we train one translator $g_0(h) = h + W_2\,\mathrm{ReLU}(W_1 h)$ with $W_1 \in \mathbb{R}^{500\times 3584}$ and $W_2 \in \mathbb{R}^{3584\times 500}$, placed before the frozen final normalization and unembedding. It reads h_0, the hidden state entering the first LLM layer, which at visual positions is the output of the visual projector. Only its $\sim$3.6M parameters are trained, about 0.05% of the backbone, and W_2 is initialized to small values so that g_0 starts near the identity. The translator is supervised by the KL divergence to the model's own final-layer distribution at the same token, over visual positions only (Section 3).

Lens training. Each lens is trained on about 87k image and question pairs drawn from counting grounding search category of Vero-600k (Sarch et al., 2026). Only the image and the question are used, since the target distribution comes from the model itself, so no annotations are needed. Training runs for $9{,}000$ AdamW steps with warmup and cosine decay, roughly one pass over the corpus, with an effective batch of 9 images, 20 warmup steps and a learning rate of $3\mathrm{e}{-4}$, and takes around 8 hours per lens on a single NVIDIA H100 GPU. The translators for the source-layer ablation (Table 8) are trained the same way at each source layer.

Target object extraction. We tokenize and POS-tag the first line of q with NLTK (Bird et al., 2009) and keep a token if it is tagged as a noun (`NN*`) and has at least one noun sense in WordNet (Miller, 1995). Consecutive kept nouns are merged into one target phrase (*fire hydrant*). We then drop a fixed list of 85 words that name no single region of an image. The list covers five closed classes: attributes predicated of an object (*color*, *size*, *shape*, and colour terms), frames of reference (*left*, *corner*, *middle*, *background*), the medium rather than its content (*image*, *photo*, *frame*, *scene*), the observer (*camera*, *viewer*, *photographer*), and bare placeholders with no referent (*object*, *thing*, *part*). The same list is used for all backbones and benchmarks. A query left with no phrase after filtering yields no crop, and the backbone answers from the original image alone.

Decoding and matching. Each cell keeps its top-K tokens with $K = 10$. The exact test lowercases a token, strips non-letter characters, and requires the result to equal one word of the target phrase. Each target is also translated once into Chinese with Google Translate, and a token matches if it equals that translation or, for translations of two or more characters, contains it or any two consecutive characters of it. The backbones decode one Chinese character per token, so a single character translation is matched by a single character token (Figure 7). If no cell matches in any grid, the tier retries once with the synonym set. WordNet first reduces the target to its base form, so *buildings* is retried as *building* (Figure 5). The set is then the lemmas of the first noun sense, followed by that sense's direct hypernyms and hyponyms, skipping multi-word lemmas and generic terms such as *entity* and *artifact*. The soft tier keeps the top 2% of cells per grid, capped at 64, before the shift.

Crop construction. By default we use 8-adjacency (link distance of one cell), one cell of padding, and $\tau = 0.2$ (Table 10). Cell boxes are mapped to pixel coordinates of I through each backbone's grid geometry. For Qwen2.5-VL, each cell covers 28×28 pixels of the resized input. For LLaVA-OneVision and InternVL3, each tile grid is placed by the tile layout and the thumbnail grid is scaled to the full image, and the boxes from both are united as in Section 3. If the strongest component alone already exceeds τ, no box is emitted for that phrase and the backbone answers from the original image.

Final inference. The prompt is a single user turn that contains the original image, then one crop per localized target phrase in the order the phrases appear in q, then the question text. No text introduces the images or crops and the question is passed unchanged. Overlapping crops are kept separate. If no target phrase yields a box, the prompt contains only the original image, which is the baseline prompt. Crops are cut from I at native resolution, and each image is then encoded by the backbone's standard processor.

Video. We uniformly sample frames at each model's standard setting (128 for LLaVA-OneVision, 120 for InternVL3, 1 fps for Qwen2.5-VL). Each frame is scored by the mean of its top five cell scores for the query phrases, and the N highest-scoring frames, kept in their original temporal order, are passed to the model in place of the full set. N is derived from the answer options (e.g., the largest option for counting questions). Questions about rendered text, such as the subtitle needles, use a text channel that matches a fixed vocabulary (*title, caption, text*). All other questions use an object channel that soft-matches the answer options.

Latency. Latency is the average wall-clock time per question from input to answer, including target extraction, the tuned lens, crop generation, and final inference. Latency result is evaluated on one single NVIDIA H100 GPU.

D Extra Results of VisLens

Tables 5, 6, and 7 break down Table 2 by subtask. Section 4.2 discusses Table 6.

V*Bench and HR-Bench. Single-target questions gain more than two-target questions in almost every case (Table 5). On V*Bench, attribute questions gain 12.1 to 13.1 points against 6.6 to 10.5 for spatial ones. On HR-Bench, FSP gains more than FCP on every backbone and resolution except InternVL3-8B at 4K. FCP includes map and chart analysis, where the target has no single location.

VisualProbe. The gains concentrate on the medium split (Table 7), where they reach 10.4 to 17.5 points and roughly double the backbone accuracy for Qwen2.5-VL-7B and InternVL3-8B. Easy questions gain 4.2 to 5.7 points, since the backbones already answer about half of them. On the hard split, Qwen2.5-VL-7B and InternVL3-8B gain 13.2 and 10.4 points, while LLaVA-OV-7B gains only 1.8.

Table 5: **Fine-grained visual search results.** Accuracy (%) of each backbone with and without VisLens on the sub-tasks of V*Bench and HR-Bench. Numbers in brackets give the number of questions in each split. Attr. = direct attributes, Spatial = relative position, FSP = fine-grained single-instance perception, FCP = fine-grained cross-instance perception. Subscripts in green denote the gain over the backbone.

Model	V*Bench			HR-Bench 4K			HR-Bench 8K		
	Attr. (115)	Spatial (76)	Overall (191)	FSP (400)	FCP (400)	Overall (800)	FSP (400)	FCP (400)	Overall (800)
LLaVA-OV-7B	78.3	65.8	73.3	73.0	54.5	63.8	65.0	52.0	58.5
w/ **VisLens**	90.4$_{12.1}$	72.4$_{6.6}$	83.2$_{9.9}$	85.0$_{12.0}$	59.8$_{5.3}$	72.4$_{8.6}$	80.0$_{15.0}$	59.2$_{7.2}$	69.6$_{11.1}$
Qwen2.5-VL-7B	67.0	61.8	64.9	71.7	52.5	62.1	61.7	51.2	56.5
w/ **VisLens**	79.1$_{12.1}$	72.3$_{10.5}$	76.4$_{11.5}$	85.8$_{14.1}$	60.6$_{8.1}$	73.2$_{11.1}$	75.5$_{13.8}$	61.8$_{10.6}$	68.6$_{11.1}$
InternVL3-8B	70.4	73.7	71.7	80.5	59.2	69.9	68.2	54.8	61.5
w/ **VisLens**	83.5$_{13.1}$	84.2$_{10.5}$	83.8$_{12.1}$	89.5$_{9.0}$	69.3$_{10.1}$	79.4$_{9.5}$	85.5$_{17.3}$	56.5$_{1.7}$	71.0$_{9.5}$

Table 6: **MME-RealWorld-Lite results by category.** Accuracy (%) of each backbone with and without VisLens. n is the number of questions in each category. Subscripts in green (red) denote the gain (drop) over the backbone.

Category	n	LLaVA-OV-7B		Qwen2.5-VL-7B		InternVL3-8B	
		Base	w/ **VisLens**	Base	w/ **VisLens**	Base	w/ **VisLens**
Perception							
Autonomous Driving	350	41.4	46.6$_{5.2}$	30.3	33.1$_{2.8}$	36.6	42.3$_{5.7}$
Diagram & Table	100	60.0	60.0$_{0.0}$	77.0	81.0$_{4.0}$	72.0	79.0$_{7.0}$
Monitoring	319	32.9	32.6$_{0.3}$	25.4	30.4$_{5.0}$	32.3	33.5$_{1.2}$
OCR (complex context)	250	82.4	83.2$_{0.8}$	80.8	85.2$_{4.4}$	84.0	88.4$_{4.4}$
Remote Sensing	150	49.3	48.7$_{0.6}$	31.3	31.3$_{0.0}$	50.7	46.7$_{4.0}$
Perception	1169	50.5	52.0$_{1.5}$	43.9	47.4$_{3.5}$	50.4	53.5$_{3.1}$
Reasoning							
Autonomous Driving	400	34.2	39.2$_{5.0}$	23.2	27.0$_{3.8}$	38.0	39.5$_{1.5}$
Diagram & Table	100	45.0	43.0$_{2.0}$	65.0	64.0$_{1.0}$	48.0	51.0$_{3.0}$
Monitoring	150	41.3	48.0$_{6.7}$	26.0	26.7$_{0.7}$	34.0	35.3$_{1.3}$
OCR (complex context)	100	71.0	72.0$_{1.0}$	66.0	68.0$_{2.0}$	71.0	68.0$_{3.0}$
Reasoning	750	42.0	45.9$_{3.9}$	35.1	37.3$_{2.2}$	42.9	44.0$_{1.1}$
Overall	1919	47.2	49.6$_{2.4}$	40.4	43.5$_{3.1}$	47.5	49.8$_{2.3}$

Table 7: **VisualProbe results by difficulty.** Accuracy (%) of each backbone with and without VisLens, following the Mini-o3 evaluation protocol with Qwen3-32B as the judge. n is the number of questions in each split. Subscripts in green denote the gain over the backbone.

		LLaVA-OV-7B		Qwen2.5-VL-7B		InternVL3-8B	
Split	n	Base	w/ **VisLens**	Base	w/ **VisLens**	Base	w/ **VisLens**
Easy	141	56.0	$61.7_{5.7}$	51.8	$56.0_{4.2}$	56.7	$61.0_{4.3}$
Medium	268	16.8	$27.2_{10.4}$	16.4	$31.7_{15.3}$	15.7	$33.2_{17.5}$
Hard	106	20.8	$22.6_{1.8}$	15.1	$28.3_{13.2}$	13.2	$23.6_{10.4}$
Overall	515	28.3	$35.7_{7.4}$	25.8	$37.7_{11.9}$	26.4	$38.8_{12.4}$

Table 8: **Source-layer ablation of the tuned lens** on Qwen2.5-VL-7B. Localization metrics are computed on small COCO objects, and the last column is V*Bench accuracy. The final-layer row decodes h_L directly after a full forward pass. Bold marks the best translated readout. Higher is better.

SOURCE LAYER	Token detection	Crop inclusion	GT coverage	V*Bench
Pre-layer 1 (Ours)	0.567	**0.390**	**0.641**	76.4
Layer 1	**0.571**	0.385	0.635	76.4
Layer 8	0.562	0.387	0.636	75.9
Layer 15	0.569	0.384	0.631	74.9
Layer 22	0.570	0.386	0.632	75.9
Final layer	0.582	0.393	0.653	78.0

D.1 Ablation: Source Layer of the Tuned Lens

Setup. We sweep the source layer ℓ of the tuned lens on Qwen2.5-VL-7B, using a held-out subset of COCO (Lin et al., 2014) objects that cover 0.01% to 1.5% of the image area. The subset is disjoint from all evaluation benchmarks, so the choice of ℓ is not tuned on reported data. We compare $\ell = 0$, the projector output, with several deeper LLM layers, and add a final-layer reference that applies the same decoding and matching to h_L after a full forward pass. This reference is the teacher of the tuned lens and serves as an upper reference for the translated readouts. We report three localization metrics. Token detection is the fraction of objects whose target word or a synonym is decoded in an overlapping cell. Crop inclusion is the fraction whose crop fully contains the ground-truth box. GT coverage is the average fraction of the ground-truth box covered by the crop.

Results. Localization is nearly flat across source layers (Table 8). The spread among translated readouts is at most 0.010 per metric with no monotonic trend in depth, and V*Bench accuracy varies by 1.5 points without a trend. The final-layer reference remains the strongest, but only by at most 0.015 in localization and 1.6 points on V*Bench, and it needs a full forward pass before cropping. The projector output at $\ell = 0$, taken before any LLM layer, is already competitive with every deeper layer. We therefore use $\ell = 0$ as the default. This supports the central design choice of VisLens, since the information needed to localize small objects is already present in the visual tokens as they enter the LLM.

D.2 Ablation: Crop Construction

Matching tier. Table 9 isolates the matching stage and holds the readout, clustering, and merging fixed. HARD keeps only the exact and synonym tiers, SOFT uses embedding-space scoring alone, and HYBRID is the full cascade of Section 3. Surface matching alone already recovers most of the gain ($+8.5$ to $+9.4$ points on V*Bench, HR-Bench, and VisualProbe), since for most queries the tuned-lens readout names the target in plain vocabulary. Embedding-space scoring is stronger, and its margin over surface matching widens with resolution ($+0.6$, $+1.5$, and $+2.8$ points on V*Bench, HR-Bench 4K, and HR-Bench 8K). This is consistent with the target's evidence surviving in the

state but falling below the top-K cutoff as the target shrinks relative to the grid. The margin of the cascade over soft alone moves in the opposite direction ($+1.5$, $+1.1$, and $+0.7$), so the two tiers are complementary rather than redundant. Only on VisualProbe does soft alone edge the cascade, by 0.3 points, which amounts to about two questions at $n = 515$.

Table 9: **Ablation on the token-matching stage.** All variants share the Qwen2.5-VL-7B backbone and differ only in how query phrases are matched to visual cells. *Baseline* is the backbone without visual search. Subscripts in green denote the gain over the baseline.

Variant	**V*Bench** (n=191)	**HR4K** (n=800)	**HR8K** (n=800)	**MME-RW-Lite** (n=1919)	**VisualProbe** (n=515)
Baseline	64.9	62.1	56.5	40.4	25.8
VisLens-hard	$74.3_{9.4}$	$70.6_{8.5}$	$65.1_{8.6}$	$43.0_{2.6}$	$34.8_{9.0}$
VisLens-soft	$74.9_{10.0}$	$72.1_{10.0}$	$67.9_{11.4}$	$43.3_{2.9}$	$38.0_{12.2}$
VisLens-hybrid (Ours)	$\mathbf{76.4}_{11.5}$	$\mathbf{73.2}_{11.1}$	$\mathbf{68.6}_{12.1}$	$\mathbf{43.5}_{3.1}$	$37.7_{11.9}$

Crop hyperparameters. The same pattern appears in the crop-level ablations. A small link distance works best, showing that only nearby components should be merged. The area cap $\tau = 0.2$ gives the best balance between context and distractors: smaller crops can miss useful surrounding evidence, while larger crops dilute the target again. Minimal cell padding is also sufficient, suggesting that VisLens benefits most from tight object-centered crops rather than broad scene crops.

Table 10: **Ablation on crop construction.** Accuracy (%) on V*Bench with the Qwen2.5-VL-7B backbone. Each block varies one hyperparameter while the other two stay at their defaults. * marks the default, and bold marks the best value in each block.

AREA CAP τ		**CELL PADDING**		**LINK DISTANCE**	
τ	Acc.	cells	Acc.	gap	Acc.
0.1	70.2	1*	**76.4**	1*	**76.4**
0.2*	**76.4**	2	**76.4**	2	75.4
0.3	74.3	3	73.8	3	73.3
0.4	73.8	4	73.8	4	73.3

E Extra Qualitative Results

Figure 7 adds four successful cases. In the umbrella and cup questions, the readout contains the answer as well as the target (`blue` and `paper`). In the license plate and trash bin questions, the answer is text and VisLens succeeds by matching the object that carries it (`bus` and `tour`, and a Chinese token for *trash*). The crop then makes the characters legible, which the full image does not. Several readouts name the target only as a Chinese token, reflecting the bilingual vocabulary of the LLM backbone.

E.1 Limitations of VisLens

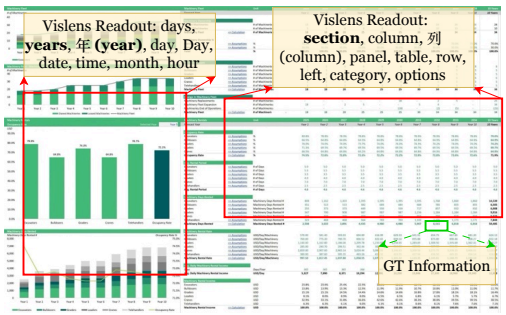
VisLens relies on local semantics. Each visual cell is decoded into tokens, and a crop forms where these tokens match the target. This works well when the target is an object or the part of an object whose name can be read from nearby cells, including when the answer is text on that object. However, it is less reliable when the queried concept has no single location.

Diagrams and tables are a typical case. A high-resolution chart is split into many cells, and each cell holds only a fragment of the content, such as a digit, a letter, or part of an arrow or line. The tuned lens then decodes these fragments rather than the concept the question asks about, so no cell matches the target (Figure 8). Aerial views fail differently, since many near-identical cells match



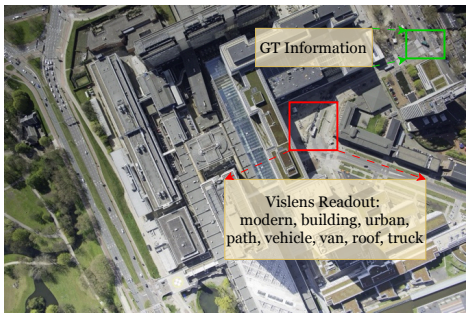
Figure 7: **Qualitative results of successful cases.** For each extracted target, the tuned lens top-K tokens at the matched region and the resulting crop (red). The backbone answers all questions wrongly, and VisLens corrects them.

the same noun and no single crop isolates the target. Both cases correspond to the weakest domains of MME-RealWorld-Lite (Table 6). VisLens also depends on the query that names the target with a word the lens decodes, as the boom case in Section 5.2 shows. Handling these inputs would require grouping cells by structure or matching relations between them, beyond the independent per-cell matching used now.



Q: In which year does the value of 'Machinery Days Rented' first exceed 6,000, according to the 'Machinery Days Rented' section?
A. 2030 B. 2031 C. 2032 D. 2033 E. The image does not feature the number.

Target Object: Machinery Days, Section. vear ? **No Valid Information**



Q: Where is the green truck located in the picture?
A. In the upper left B. In the upper right C. In the lower left D. In the lower right E. This image doesn't feature the position.

Target Object: Truck ? **No Valid Information**

Figure 8: **Failure cases on a table and an aerial image.** For each question, the tuned-lens tokens at the crop (red) and the ground-truth region (green). The matched words *year* and *section* recur across the table (left), and *truck* is decoded on multiple regions (right), so neither crop contains the target.