DIAGNOSING THE REFERENCE GAP: WHAT VISUAL SCAFFOLDS CAN AND CANNOT FIX IN MULTIMODAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models can recognize visual content yet struggle to reason reliably over it across multiple steps. In counting, a model may see all objects but lose track of which ones it has counted. We argue that such failures, commonly attributed to perception, reveal a reference gap: the failure to bind language to the relevant visual entities, regions, or locations. Motivated by the theory of visual routines, we test whether external references such as numbering can narrow this gap. We introduce ReGap-Bench, a paired benchmark of five visual reasoning tasks in which the scene, question, and answer are fixed and only the reference varies. Results show that task-matched references improve accuracy by up to 25%, while mismatched references provide little benefit or even hurt performance. These findings identify the reference gap as a bottleneck distinct from visual perception and show that constructing task-appropriate visual references can serve as a training-free visual skill for improving multimodal reasoning.

1 INTRODUCTION

Multimodal large language models (MLLMs) have become increasingly capable at recognizing objects, shapes, and spatial relations in single-step perceptual tasks (Dong et al., 2026; Team et al., 2026), yet they often struggle to maintain and use that information across multiple reasoning steps. For example, a model may detect every object in a scene but still miscount them. Much of the field's response to such failures has focused on perception, pursuing higher-resolution encoders and finer-grained visual features (Tong et al., 2024b). However, the missing capability may instead lie between perception and reasoning: models lack a stable mechanism for referring to objects, regions, or locations while reasoning. Figure 1 shows an example of a counting problem. The model refers to shapes by vague locations such as "on the left side" and counts one ellipse twice; when every shape carries a number, it lists the ellipses by label and answers correctly.

Following Lu et al. (2026), we call this the *reference gap*: the failure to maintain stable bindings between language-based reasoning and the relevant visual entities, regions, or locations. Recent work addresses it either by training a model to produce coordinates within its reasoning (Lu et al., 2026), or by drawing lines on the image and instructing the model to scan along them (Izadi et al., 2025). Two questions remain open. First, neither isolates the effect of the reference itself. Lu et al. (2026) compare a newly trained model with other models, so part of the gain may come from training data or architecture. Izadi et al. (2025) varies both the image and the prompt together, so it is unclear if MLLMs can perform systematic scanning of the image without additional instructions from the prompt. Second, each applies the same reference to every task, although tasks might need different visual operations. It thus remains unclear which reference a task needs, and whether any added structure would help equally.

To design such a diagnostic framework, we identify visual routines used for complex visual reasoning, such as marking visited items, activating bounded regions, and tracing boundaries (Ullman, 1984). We hypothesize that MLLMs lack internal reference frames for reliable execution of these visual routines, and that external references such as numbering, shading, or grid lines can support them, with each reference helping most on tasks that demand its visual routines. To test this hypothesis, we introduce **ReGap-Bench**, a controlled benchmark of paired reference interventions across

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

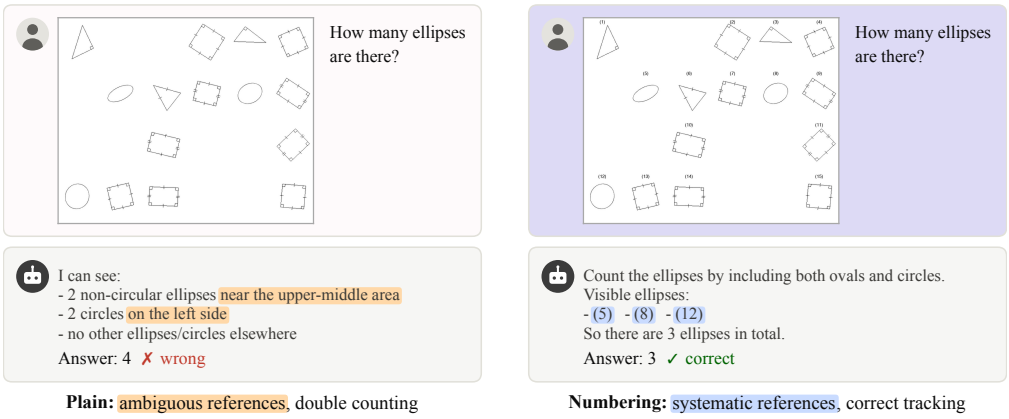


Figure 1: The same question to GPT-5.4 mini about the same scene, without and with numbering reference. In the plain setting (left), the model refers to shapes by vague locations and double counts. With numbering (right), it lists the ellipses by label and answers correctly.

five tasks: shape counting, angle ranking, size ranking, clock ranking, and transformation prediction. Each underlying scene is rendered with and without explicit visual references, while its task-relevant objects, geometry, question, and gold answer remain fixed.

Results show that task-matched references improve accuracy on all five tasks. Numbering yields gains of up to 17.5% on shape counting and 25% clock ranking; shading improves angle and size ranking; and grids improve size ranking and transformation prediction. Next, we show that the gains depend on matching the reference to the task, not on adding marks. Additionally, model scale across six Qwen3-VL models from 2B to 235B does not remove the reference gap. Building on these findings, we introduce a **training-free visual skill** that enables an agent to dynamically identify the required visual routine and select the corresponding reference scaffold. Evaluated on CartoMapQA (Ung et al., 2025), this skill enables autonomous reference selection and improves complex map understanding without finetuning. We summarize our contributions as follows:

- We introduce ReGap-Bench, a paired evaluation framework that measures the reference gap. It spans five visual reasoning tasks and three reference systems, with scene semantics, questions, and answers held fixed across reference conditions.
- We show that task-matched references improve all evaluated models on all five tasks. Mismatched, incomplete, and model-scale analyses provide evidence that the gains come from supporting the visual routine a task requires, and that they persist from 2B to 235B parameters.
- We introduce a training-free visual skill that enables an agent to identify the visual routine required by a task and dynamically construct a corresponding visual reference at inference time.

2 Diagnosing the Reference Gap with Visual Interventions

2.1 The reference gap

We use the reference gap to describe the failure to maintain a stable link between language and the relevant visual entity, region, or location that the language corresponds to. Models might perceive the relevant visual information, yet struggle to identify, revisit, or distinguish from other visual elements as reasoning proceeds. Consider shape counting. A model may correctly recognize individual triangles when asked about them separately yet still miscount them (see preliminary analysis in Appendix A). The model needs to repeatedly select an object, determine whether it belongs to the target class, keep track of counted objects, and then move to the next. Without a stable internal reference, it can lose track of which objects have been counted, resulting in omissions or duplicates. The reference gap is therefore distinct from a perception failure, and it should matter most in visual reasoning that spans multiple steps.

Most benchmarks do not reveal whether the model failed to perceive the relevant content or failed to maintain reference to it during reasoning. To address this, we diagnose the reference gap by testing

whether performance changes when the underlying visual problem is held fixed but the model is given explicitly external reference to refer to its visual elements.

2.2 External references for visual routines

To choose useful visual references, we draw on Ullman's theory of visual routines (Ullman, 1984), which describes complex visual reasoning as a sequence of elementary operations such as shifting the processing focus, bounded activation, boundary tracing, indexing, and marking. We propose three types of visual references that externalize intermediate visual states to support these routines.

Numbering. Numbering supports marking and indexing. It turns description-dependent visual elements into addressable entities. For instance, instead of using ambiguous phrases like "the upper-left triangle", the model can refer to "shape 7". This makes it easier to find an item again and to track which items have already been processed.

Shading. Shading supports bounded activation by making a region explicit. For example, shading the interior of an angle makes the angular region itself visible. Filling a shape similarly emphasizes its bounded area rather than only its outline.

Grid lines. Grids provide spatial marks that can support systematic focus shifts and boundary tracing. They divide the image into regular grid intervals and can provide a reference for position, width, height, displacement, and alignment.

These scaffolds provide different kinds of reference information. Therefore, we do not expect every scaffold to help every task. Our main hypothesis is that: a reference scaffold helps most when the type of reference it provides matches the visual routines that the reasoning process needs.

2.3 Paired reference-intervention design

To test whether a reference gap exists, we design a paired reference intervention in which we hold the problem fixed and intervene only on the external reference available to the model. For each problem, we create a plain image and one or more versions with added references, all generated from the same underlying scene.

Formally, a problem is a triplet (s, q, y), where s is the scene specifying the objects, geometry, and relations; q is the question; and $y = f(s, q)$ is the ground truth determined by the scene and question.

Let $\mathcal{C}$ be a set of reference conditions including the plain condition c_0, and let $x_c = R(s, c)$ denote the image obtained by rendering scene s under condition $c \in \mathcal{C}$. Every reference condition satisfies three properties: (i) *content preservation*: every element of s appears in x_c with the same shape, size, and position as in x_{c_0}, and the reference only adds numbers, changes fill color, or adds background structure; (ii) *answer invariance*: the target $y = f(s, q)$ is identical for all $c \in \mathcal{C}$ as it only depends on s and q; and (iii) *no answer leakage*: the reference alone does not determine y, since numbers and colors are assigned to all elements independently of their class. Each problem therefore yields a set of images $\{x_c\}_{c\in\mathcal{C}}$ that differ only in the reference available to the model.

Given a model M, let $\hat{y}_c = M(x_c, q)$ denote its prediction under condition c, and let $a_c = \mathbf{1}[\hat{y}_c = y]$ denote its correctness. For a reference condition $c \neq c_0$, the effect of adding the reference is $\delta(c) = a_c - a_{c_0} \in \{-1, 0, 1\}$. For a dataset $\mathcal{D} = \{(s_i, q_i, y_i)\}_{i=1}^N$ rendered under condition c, the average reference effect is:

$$\Delta(c) = \frac{1}{N}\sum_{i=1}^{N} \delta_i(c) = \mathrm{Acc}(c) - \mathrm{Acc}(c_0), \qquad \mathrm{Acc}(c) = \frac{1}{N}\sum_{i=1}^{N} a_{i,c}. \tag{1}$$

A positive $\Delta(c)$ suggests a reference gap: the model is able to solve more of the same problems once the reference is made explicit. Since adding marks also introduces additional visual information, we further test in Section 4.2 whether the gains come from the reference itself. A close-to-zero $\Delta(c)$ means the added reference brings no benefit, either because the model already maintains reference on its own or because its errors lie somewhere else, such as in perception. A negative $\Delta(c)$ means the reference interferes with the task, for example by adding clutter to the scene.

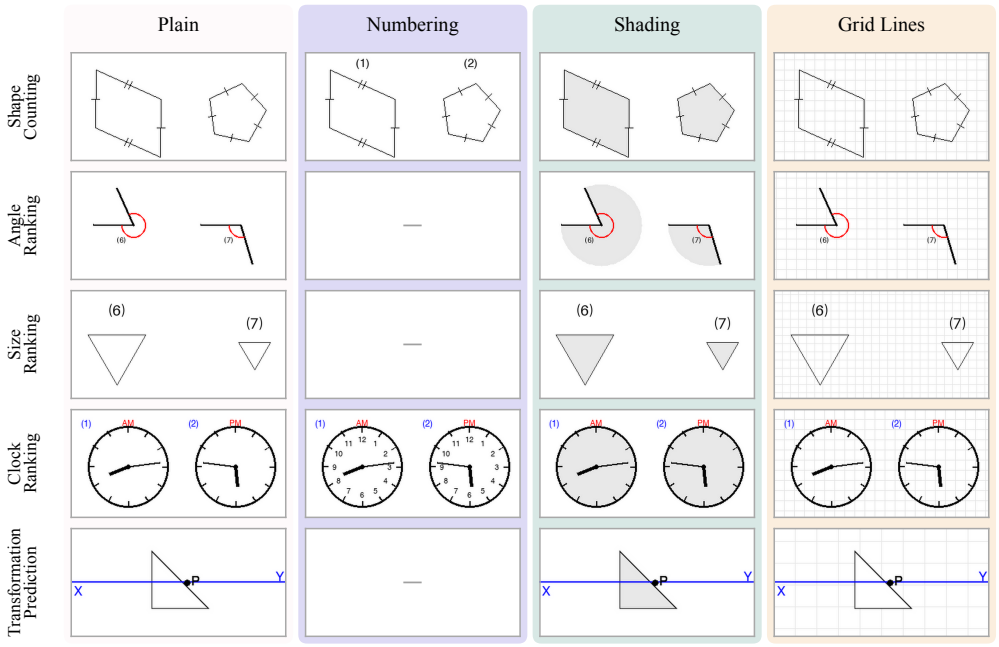


Figure 2: Example ReGap-Bench scenes. Each row shows a crop of one scene under the plain and each reference condition. The scene content, question, and answer are identical across a row. Numbering is not applied to angle ranking, size ranking, or transformation prediction.

3 REGAP-BENCH: A REFERENCE-INTERVENTION BENCHMARK

We introduce **ReGap-Bench**, applying the paired design of Section 2.3 to five tasks: shape counting, angle ranking, size ranking, clock ranking, and transformation prediction. Each task has 200 scenes, each rendered to different reference conditions, as shown in Figure 2. A reference only adds numbers, fills interiors with light gray, or adds faint grid lines. It does not move, remove, or cover any original (*content preservation*), and it treats all elements alike, so it does not reveal the answer (*no answer leakage*). Appendix B shows generation details and dataset statistics.

3.1 TASKS

We design five tasks that share a common requirement: the model must shift its focus across different parts of an image and repeatedly bind reasoning steps to particular visual elements. For each task, we identify the visual routine it relies on most and the reference scaffolds we expect to support that routine, which we call *matched*.

Shape counting. The scene contains outlined shapes such as triangles, quadrilaterals, and pentagons, arranged in rows and columns at random orientations. We annotate the shapes with standard geometry marks, such as tick marks on equal sides and small squares at right angles, to make each shape's class unambiguous. The question asks for the count of a specific shape. The model must select each shape, classify it, and record it as counted without skipping or repeating any. The task therefore relies on indexing and marking. Numbering could be a matched reference because it assigns each object a stable address that can be used throughout the counting process.

Angle ranking. The scene contains several angles, each formed by two rays from a vertex, and marked by a red arc. The model is tasked to rank the angles by size, from smallest to largest, or the reverse. The main difficulty is keeping the correct angular region active: looking at the wrong side of the rays turns an angle into its conjugate angle. Shading the angular wedge could be helpful in making the region being compared explicit and thus supports bounded activation.

Size ranking. The scene contains several shapes of one class. The model is asked to rank the shapes according to their visual size. This requires estimating the size of each shape and maintain these

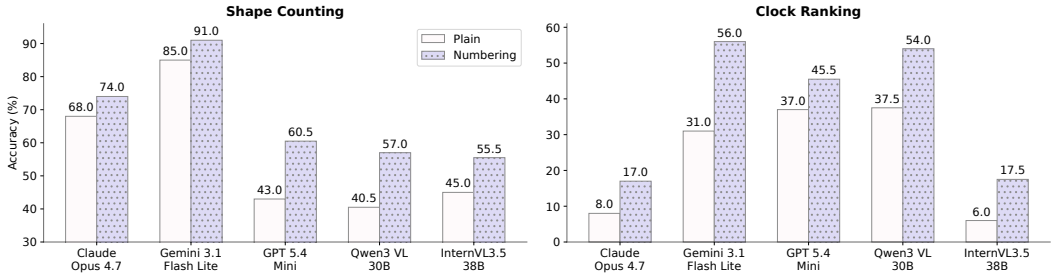


Figure 3: Numbering improves performance across all models on shape counting and clock ranking.

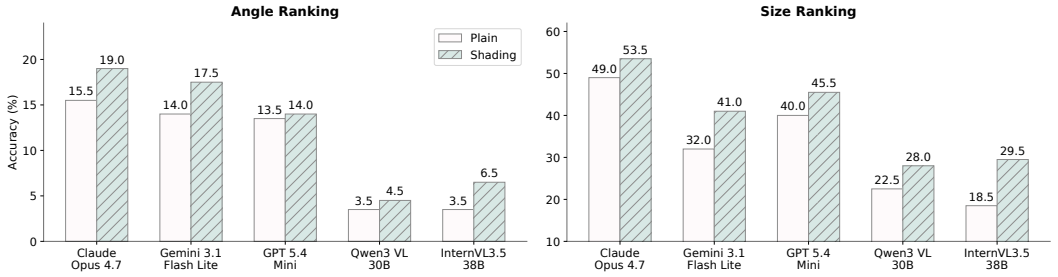


Figure 4: Shading improves across all evaluated models on both angle ranking and size ranking.

estimates while shifting the focus to other shapes. We expect two references to help: shading makes each shape's area explicit as a bounded region, and grid lines provide a common scale for comparing width and height across locations.

Clock ranking. The scene contains several analog clocks, each with twelve tick marks, an hour hand, a minute hand, and an "AM" or "PM" label. The task is to rank the clocks from earliest to latest, or the reverse. The main reference challenge is mapping hand positions onto a stable circular reference frame. We expect dial numbering to help, because it marks positions on that frame directly.

Transformation prediction. Given a source figure and a chain of transformation steps, the model needs to determine which of the four given options is the final figure. Transformation operations include translation, reflection, rotation, and dilation. To solve the task, the model must trace the shape through each step and shift focus across panels to compare final positions. We expect grid lines to help, because they provide give a common frame for position and scale.

3.2 Evaluation Setup

We evaluate five MLLMs from both proprietary and open-source families: Claude-Opus-4.7 (Anthropic, 2026), Gemini-3.1-Flash-Lite (Gemini Team, 2026), GPT-5.4-mini (OpenAI, 2026), Qwen3-VL-30B-A3B-Thinking (Team, 2025), and InternVL3.5-38B (Wang et al., 2025). In Section 4, we refer to models by their family names (e.g., Gemini). To study the effect of model scale, we additionally evaluate six Qwen3-VL Thinking models, from 2B to 235B parameters[1]. Prompt template is shown in Appendix C.1. Our main metric is exact match. For ranking tasks, we additionally report pairwise accuracy, the share of pairs that the predicted ranking puts in the correct order, which gives partial credit. We compute it over valid rankings.

4 Experiment

4.1 Task-matched references improve visual reasoning

We begin with the operation-matched setting: for each reference, we test two tasks whose required visual routine could be supported by that scaffold.

[1]https://huggingface.co/collections/Qwen/qwen3-vl

Numbering supports indexing and marking. Figure 3 shows that numbering consistently improves performance, with a performance gain of 6.0% to 17.5% on shape counting, and 8.5% to 25.0% on clock ranking. On shape counting, models with low accuracy in the plain condition gain the most: GPT and Qwen gain about 17%, while Gemini and Claude gain 6%. To examine how numbering changes the counting process, we manually label each model response by the reference strategy it uses: (i) *systematic references*, which use numbered identifiers (e.g., "shape (7)") and a explicit row- or column-based scheme (e.g., "Top row"); and (ii) *ambiguous references*, which use vague spatial descriptions (e.g., "the upper-left one") or provide no clear reference scheme. On plain images, Gemini frequently constructs its own reference system: 52.0% of its responses assign numbers to the shapes, and 31.5% explicitly scan them by rows or columns. In contrast, GPT does so much less often: only 15.5% of its responses use either *systematic references* strategy, and the other 84.5% use *ambiguous references*. With numbering, 100% of Gemini's and 86.0% of GPT's responses refer to the printed numbers. This suggests that **numbering supplies stable symbolic addresses that are particularly useful for models that lack their own reference system**.

For clock ranking, we extract the time each model stated for each clock from its response and count a clock as correctly-read if it is within 5 minutes of the ground truth. Among clocks with a stated time, numbering raises this rate from 10.6% to 28.6% for Claude, 58.2% to 82.5% for Gemini, 67.5% to 75.6% for Qwen3-VL, and 2.9% to 17.4% for InternVL. Thus, **dial numbers improve these models' ability to read individual clocks, an intermediate step required for correct ranking**. GPT behaves slightly differently: when it states a time, it is almost accurate (90.3% plain vs. 90.8% with numbering), but on plain images it gives no time for 34% of the clocks, and this propotion is reduced to 12% with numbering. For GPT, therefore, **numbering primarily improves the completeness of the intermediate clock-reading process rather than the accuracy of readings**.

Shading supports bounded activation. Figure 4 shows the effects of shading on angle ranking and size ranking. Shading improves size ranking from 4.0% to 11.0%, while the gains on angle ranking are relatively small. As exact-match is a very strict metric for this task, and plain accuracy is all below 16%, we additionally study the pairwise accuracy (Appendix D). Pairwise accuracy improves with shading for four of the five models, including from 76.4% to 85.2% for Gemini and from 59.7% to 68.0% for Qwen3-VL. Thus, **shading improves the overall ordering of the angles significantly**.

Table 1: Size-ranking pairwise accuracy (%, plain → shading) by scale-factor difference.

	Scale-factor difference		
Model	**0.05–0.10**	**0.10–0.20**	**≥ 0.20**
Claude	82.2 → 84.3	95.6 → 96.6	99.8 → 99.9
Gemini	79.3 → 82.4	91.8 → 93.1	99.3 → 99.5
GPT	80.0 → 80.8	93.4 → 94.4	99.5 → 99.6
Qwen3-VL	71.6 → 74.0	84.3 → 85.7	97.6 → 98.2
InternVL	67.5 → 73.6	84.2 → 86.8	97.5 → 98.7

On size ranking, pairwise accuracy is already above 91% for every model on plain images, and the remaining errors lie almost entirely between shapes of similar size: pairs whose scales differ by less than 0.10 are ordered correctly only 67.5% to 82.2% of the time, compared with at least 97.5% for pairs that differ by 0.20 or more (Table 1). Shading helps most on those close pairs, suggesting that *the exact-matching gains primarily come from resolving swaps between similarly sized shapes*.

To study how shading helps angle ranking, we measure one key error that it likely prevents: reading the wrong side of rays, which turns an angle θ, into its conjugate angle, $360^\circ - \theta$. For instance, Claude reads one angle as about 10° on the plain image and as about 350° with shading, where the true value is 351°. As models do not always state a degree value for each angle, we measure this error from the final rankings. We take each angle below 90° or above 270°, for which the two readings are far apart, and check whether the model placed it closer to where θ belongs or to where $360^\circ - \theta$ would belong. Detailed implementation can be found in Appendix C.4. With shading, it almost halves this rate for Claude (3.3% to 1.7%), Gemini (18.4% to 8.5%), and InternVL (23.9% to 13.1%). This shows that **shading provides a bounded visual region that helps models maintain the intended side of the rays during measurement**. GPT-5.4-mini shows no such reduction (13.3% vs. 13.8%), consistent with its nearly unchanged pairwise accuracy under shading.

Grids support boundary tracing and shifting focus. Figure 5 shows that grid lines consistently improve performance, with a performance gain of 2.0% to 8.0% on size ranking, and 2.0% to 5.5% on transformation prediction. Notably, in size ranking, grid lines yields smaller gains than shading in four models. One possible explanation is that using grids requires several additional operations:

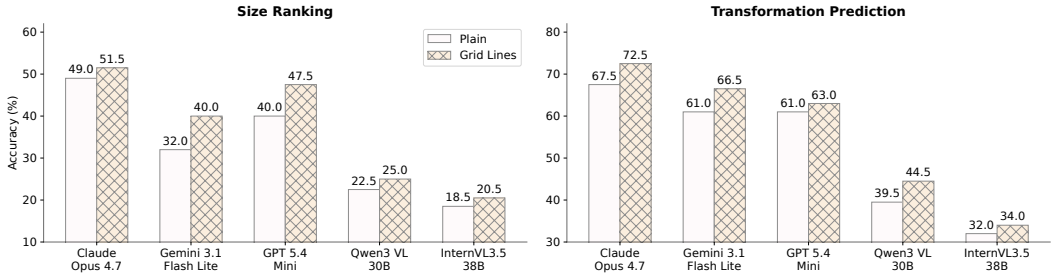


Figure 5: Grid lines improve all evaluated models on size ranking and transformation prediction.

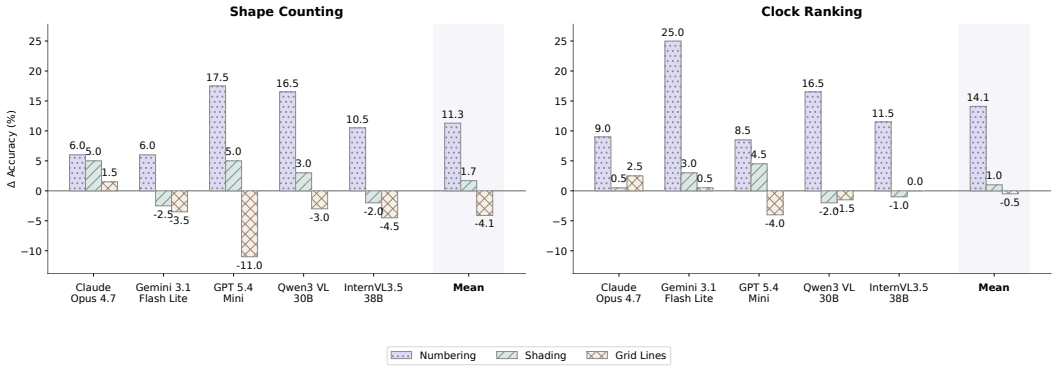


Figure 6: Operation-matched numbering produces consistently larger gains than shading or grid lines. Shading provides only small or inconsistent gains, while grid lines are often neutral or detrimental, suggesting that visual references help when they support the routines required by the task.

the model must measure each shape in grid cells, keep each reading, and then compare the readings. These operations themselves depend on maintaining stable references. On the other hand, shading makes the region of each shape directly visible.

To examine whether models indeed use grids for measurement or position tracking, we use an LLM judge to classify the model responses by the reference strategy : (i) *systematic references* that give at least one position, distance, or size in grid cells or as grid coordinates; (ii) *ambiguous references* that describe position and size only in general terms, which is detailed in Appendix C.5. Gemini uses *systematic references* in 51% of its responses in transformation prediction, usually by writing the vertices as grid coordinates and following them through each step (e.g., "The vertices are roughly at $(-1, 0)$, $(-1, 2)$, and $(0, 0)$ relative to P"). In contrast, GPT never adopts this strategy: it relies on qualitative descriptions such as "left of P", "half the size", or "closer to P", regardless of plain or grid condition. This difference is consistent with the larger improvement from grids for Gemini than for GPT (5.5% vs. 2.0%), suggesting that **the benefit of a reference scaffold depends not only on its presence, but also on whether the model incorporates it into its reasoning process**.

4.2 Analysis on Matched Versus Mismatched References

The positive scaffold results establish that each reference method can help in matched settings. However, it is unclear whether these gains simply come from adding more visual marks. We test this by applying numbering, shading, and grid lines to the same shape counting and clock ranking scenes. If improvements were caused merely by additional visual information, then numbering, shading, and grid lines should help similarly. If the reference gap exists, then the operation-matched scaffold should produce the largest gain. Figure 6 supports the second explanation. In shape counting, numbering improves accuracy by 11.3% on average, compared to only 1.7% for shading, while grid lines reduce accuracy by 4.1%. The same separation appears in clock ranking. These results show that added visual structure is not always beneficial: a scaffold can be neutral or harmful when it does not provide the kind of reference the task needs. **The effectiveness of a scaffold therefore depends on its alignment with the visual routine needed for solving the task.**

4.3 Analysis on Incomplete References

Next, we test whether references help because they provide complete reference structure, or whether partial reference is sufficient. We test this on clock ranking by varying how many dial numerals are shown while keeping the clocks themselves fixed. The four conditions contain zero, two, four, or all twelve numerals. If numbering helps because it provides stable addresses, then a more complete reference system should be more useful than a sparse one. Figure 7 confirms this, mean accuracy rises from 23.9% with no numerals to 25.3% with two, 31.8% with four, and 38.0% with all twelve. The trend is non-decreasing for every evaluated model. Sparse numbering therefore provides only a small benefit, whereas a complete reference system produces a substantially larger gain. This result is consistent with a reference-based mechanism: a scaffold is most useful when it supplies stable addresses that the model must repeatedly identify during reasoning. **Partial references can still help, but they leave a larger portion of the task dependent on the model's own spatial descriptions or implicit object bindings.**

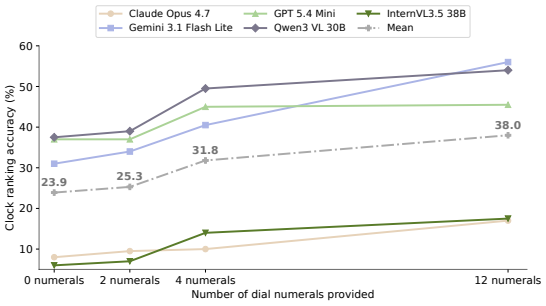


Figure 7: Performance improves as dial numbering becomes more complete.

4.4 Analysis of Model Scale

The final question is whether the reference gap is primarily a limitation of small models. We evaluate six Qwen3-VL Thinking models ranging from 2B to 235B parameters on shape counting. Figure 8 shows a clear overall increase in accuracy with model scale. However, the advantage of numbering remains at every scale: the gain ranges from 9.0% to 23.5% and does not decrease monotonically as models become larger. Thus, **a larger model size improves the task performance but does not eliminate the benefit of an explicit reference system**.

5 Task-matched references as training-free visual skills

In Section 4, we show that models that lack an internal reference system can benefit from task-matched references *without any training*. One implication is that agents can add these references on their own using a visual skill when a task requires such reference. Unlike most skills that are text-only, we package the three references as a visual skill that can refer to visual examples and overlay the reference onto the image similarly.

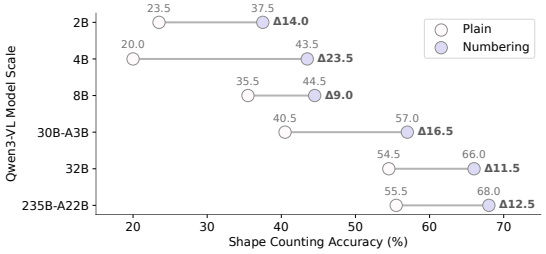


Figure 8: Scaling model size help both plain and numbering, but does not eliminate the reference gap.

We test the skill on CartoMapQA (Ung et al., 2025) that contains street-level OpenStreetMap renderings with icons for points of interest (POIs), road labels, markers, and a scale bar. We use stratified sampling to draw 100 tasks across the eight subtasks, and run the Codex GPT-5.5 agent on the same tasks with and without the skill, using the BenchFlow framework (BenchFlow Team, 2026).

The skill contains three components: First, written rules specify which reference to use for each type of visual operation: numbering for counting, listing, or matching discrete map elements, as well as tracking junctions along a route; grid lines, calibrated using the map's scale bar, for measuring distances or comparing positions; shading for isolating bounded regions; and no overlay when the answer can be obtained from a single clearly labeled element. Second, a pair of before-and-after example images demonstrating how to draw each reference. Third, a script draws the chosen reference on a copy of the map and leaves the original unchanged.

Table 2 shows that all references improve the mean score. The agent draws an overlay on 57 of the 100 tasks, and the selected references broadly align with the visual operations required by each subtask: numbering for counting, listing, and route navigation; grid lines for route-length tasks; shading for region isolation for listing tasks; and no overlay for symbol recognition and presence questions. These results suggest that **an agent can select and construct task-relevant references through a reusable skill without additional model training**.

Table 2: Visual skill performance (accuracy, %) on CartoMapQA, broken down by overlay type.

Overlay	Tasks	No skill	With skill	Δ
None	43	86.0	86.0	0.0
Numbering	41	63.7	70.1	+6.4
Grid lines	13	92.5	96.7	+4.2
Shading	3	33.3	66.7	+33.3

6 Related Work

Visual perception and reasoning in MLLMs. Benchmarks including BLINK (Fu et al., 2024), MMVP (Tong et al., 2024b), BlindTest (Rahmanzadehgervi et al., 2024), CV-Bench (Tong et al., 2024a), and V*Bench (Wu & Xie, 2024) study limitations in visual perception and reasoning. Related studies report difficulties in spatial reasoning (Wang et al., 2024; Yang et al., 2025), enumeration (Tamarapalli et al., 2025; Deng et al., 2026), and chart understanding and reasoning (Masry et al., 2025). However, they do not distinguish failures to encode visual elements from failures to maintain references during reasoning. Our paired design probes the effects of visual references by holding the underlying scene, question, and answer fixed while varying the reference system.

Grounding intermediate reasoning in visual references. Lu et al. (2026) identify the *reference gap* as a failure to ground verbal reasoning in precise visual entities and locations. Such failures can propagate across reasoning steps in dense counting, spatial deduction, and path tracing. Their approach trains models to generate points and bounding boxes within their reasoning trajectories through large-scale primitive supervision, reinforcement learning, and distillation. BLINK highlights tasks that are difficult to mediate through natural language (Fu et al., 2024). On VSI-Bench, Yang et al. (2025) find that chain-of-thought prompting, self-consistency, and tree-of-thoughts reduce Gemini-1.5 Pro Team et al. (2024)'s average performance in their evaluation, while explicitly generating cognitive maps improves its relative-distance accuracy. We examine whether references supplied in the input image can support these bindings in a fixed, off-the-shelf model, and under what conditions the model uses them.

Visual prompting and grounded reasoning. Prior work supports visual reasoning through annotations, auxiliary visual states, and modular computation. Red-circle prompting directs CLIP towards selected regions (Shtedritski et al., 2023), while Set-of-Mark supplies identifiers for segmented regions (Yang et al., 2023). Visual Sketchpad generates auxiliary drawings (Hu et al., 2024), PIVOT refines action proposals through numbered annotations (Nasiriany et al., 2024), and Draw-and-Understand trains models to interpret user-drawn prompts (Lin et al., 2025). VisAidMath evaluates mathematical reasoning with textually described visual aids (Ma et al., 2026). Other approaches compose vision modules through code (Surís et al., 2023), incorporate visual features into rationale generation (Zhang et al., 2024), or represent intermediate reasoning states as images (Meng et al., 2023). Our study systematically varying references rendered in the input image to test which reference systems support particular visual routines and whether their benefits depend on task alignment, with task-matched references implemented as reusable skills that require no model training.

7 Conclusion

We identify the *reference gap* as a distinct bottleneck in multimodal reasoning: models may perceive the relevant visual content yet fail to maintain stable references to it during multi-step reasoning. Using ReGap-Bench, we show that task-matched visual references consistently improve performance, whereas mismatched references provide little benefit or can be harmful, and that this gap persists even at larger model scales. These results suggest that improving multimodal reasoning requires mechanisms that support the visual routines needed by the task. Finally, our training-free visual skill demonstrates that an agent can dynamically construct such references at inference time, offering a simple way to improve visual reasoning without additional model training.

AI USE STATEMENT

In this work, we used generative AI tools to refine hypotheses and support qualitative analysis. We have not used generative AI tools to generate synthetic data sets, clean or reformat dataset. Lastly, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, and proving mathematical claims are not applicable to this work.

Additionally, we used generative AI tools to edit software code, polish a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. For example, we carefully reviewed the AI-refined paper content and evaluated AI-assisted implementation using manually written unit tests. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Introducing Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, april 2026. Accessed: 2026-09-25.

BenchFlow Team. Benchflow: framework for rl environments for llm agents, 2026. URL https://github.com/benchflow-ai/benchflow.

Jinhong Deng, Limeng Qiao, and Guanglu Wan. Holocount: A holistic visual counting benchmark for mllms. *CoRR*, abs/2607.06420, 2026. doi: 10.48550/ARXIV.2607.06420. URL https://doi.org/10.48550/arXiv.2607.06420.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-adv: Exploring MLLM visual reasoning in referring expression tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=iEBgrepR9i.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: multimodal large language models can see but not perceive. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXIII*, volume 15081 of *Lecture Notes in Computer Science*, pp. 148–166. Springer, 2024. doi: 10.1007/978-3-031-73337-6_9. URL https://doi.org/10.1007/978-3-031-73337-6_9.

Gemini Team. Gemini 3.1 Flash-Lite: Built for intelligence at scale. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/, March 2026. Google Blog. Accessed: 2026-09-25.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/fb82011040977c7712409fbdb5456647-Abstract-Conference.html.

Amirmohammad Izadi, Mohammadali Banayeeanzade, Fatemeh Askari, Ali Rahimiakbar, Mohammad Mahdi Vahedi, Hosein Hasani, and Mahdieh Soleymani Baghshah. Visual structures help visual reasoning: Addressing the binding problem in LVLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=T52hZeT7rn.

Weifeng Lin, Xinyu Wei, Ruichuan An, Peng Gao, Bocheng Zou, Yulin Luo, Siyuan Huang, Shanghang Zhang, and Hongsheng Li. Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=bfa58H1nQ8.

Ruijie Lu, Xiaokang Chen, Xingchao Liu, Yutong Lin, Hao Li, Wen Liu, Zhewen Hao, Xi Gao, Shaoheng Nie, Yixuan Wei, et al. Thinking with visual primitives. Technical report, DeepSeek-AI, 2026. URL https://www.k-a.in/Thinking_with_Visual_Primitives.pdf.

Jingkun Ma, Runzhe Zhan, Yang Li, Di Sun, Hou Pong Chan, Lidia S. Chao, and Derek F. Wong. VisAidMath: Benchmarking visual-aided mathematical reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37057–37103, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1719. URL https://aclanthology.org/2026.acl-long.1719/.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md. Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md. Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Chartqapro: A more diverse and challenging benchmark for chart question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 19123–19151. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.978. URL https://doi.org/10.18653/v1/2025.findings-acl.978.

Fanxu Meng, Haotong Yang, Yiding Wang, and Muhan Zhang. Chain of images for intuitively reasoning. *CoRR*, abs/2311.09241, 2023. doi: 10.48550/ARXIV.2311.09241. URL https://doi.org/10.48550/arXiv.2311.09241.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. PIVOT: iterative visual prompting elicits actionable knowledge for vlms. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37321–37341. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/nasiriany24a.html.

OpenAI. Introducing GPT-5.4 mini and nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/, march 2026. Accessed: 2026-09-25.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In Minsu Cho, Ivan Laptev, Du Tran, Angela Yao, and Hongbin Zha (eds.), *Computer Vision - ACCV 2024 - 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8-12, 2024, Proceedings, Part V*, volume 15476 of *Lecture Notes in Computer Science*, pp. 293–309. Springer, 2024. doi: 10.1007/978-981-96-0917-8_17. URL https://doi.org/10.1007/978-981-96-0917-8_17.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does CLIP know about a red circle? visual prompt engineering for vlms. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11953–11963. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01101. URL https://doi.org/10.1109/ICCV51070.2023.01101.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 11854–11864. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01092. URL https://doi.org/10.1109/ICCV51070.2023.01092.

Jayant Sravan Tamarapalli, Rynaa Grover, Nilay Pande, and Sahiti Yerramilli. Countqa: How well do mllms count in the wild? *CoRR*, abs/2508.06585, 2025. doi: 10.48550/ARXIV.2508.06585. URL https://doi.org/10.48550/arXiv.2508.06585.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Qwen Team. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025. doi: 10.48550/ARXIV.2511.21631. URL https://doi.org/10.48550/arXiv.2511.21631.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9568–9578. IEEE, 2024b. doi: 10.1109/CVPR52733.2024.00914. URL https://doi.org/10.1109/CVPR52733.2024.00914.

Shimon Ullman. Visual routines. *Cognition*, 18(1–3):97–159, 1984.

Huy Quang Ung, Guillaume Habault, Yasutaka Nishimura, Hao Niu, Roberto Legaspi, Tomoki Oya, Ryoichi Kojima, Masato Taya, Chihiro Ono, Atsunori Minamikawa, and Yan Liu. Cartomapqa: A fundamental benchmark dataset evaluating vision-language models on cartographic map understanding. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '25, pp. 440–453, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720864. doi: 10.1145/3748636.3762755. URL https://doi.org/10.1145/3748636.3762755.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. Is A picture worth A thousand words? delving into spatial reasoning for vision language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/89cc5e613d34f90de90c21e996e60b30-Abstract-Conference.html.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, JingJing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025. doi: 10.48550/ARXIV.2508.18265. URL https://doi.org/10.48550/arXiv.2508.18265.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 13084–13094. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01243. URL https://doi.org/10.1109/CVPR52733.2024.01243.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 10632–10643. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00994. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=y1pPWFVfvR.

A PRELIMINARY CHECK: CAN MODELS PERCEIVE THE SHAPES?

Section 2.1 claims that a model can perceive the shapes in a scene yet still fail to count a target class. As a preliminary check of this claim, we run two simpler probes on the plain shape-counting scenes with Claude-Opus-4.7.

Finding the shapes. We ask the model how many shapes the image contains, regardless of class. It answers correctly in 193 of the 200 scenes (96.5%; Table 3). The errors grow with the number of shapes, and all six errors in scenes with at most 10 shapes overcount by exactly one.

Table 3: Counting all shapes, regardless of class (Claude-Opus-4.7, plain images).

SHAPES IN IMAGE	1–10	11–20	21–30	31–40	41–50	All
Correct	38 / 38	92 / 92	44 / 45	18 / 22	1 / 3	193 / 200

Recognizing single shapes. For each scene, we crop one shape from the plain image and ask whether it belongs to the scene's target class, with the same inclusion rules as in the counting prompt. For example, the scene question "How many parallelograms are there?" becomes "Is this shape a parallelogram?". If the scene contains the target class, we crop a random target shape (answer Yes); otherwise, we crop a random shape (answer No). The model answers correctly for 95.9%.

Summary. The model counts all shapes correctly in 96.5% of scenes and recognizes the single shape in 95.9%, but counts the target class correctly in only 72.0%. Most counting errors therefore do not come from missing or misreading individual shapes.

B REGAP-BENCH DETAILS

All scenes are generated in Python with Pillow from a fixed random seed. For each scene, we first sample a specification (the items, their geometry, and the question) and then draw every condition from that same specification, so item positions, sizes, and orientations are identical across conditions. The gold answer is computed from the specification.

B.1 DATASET STATISTICS

Table 4 summarizes the benchmark. The 1,000 scenes yield 3,800 images.

Table 4: ReGap-Bench statistics. Each task has 200 scenes. Layout is the number of rows × columns of cells. Means are given in parentheses.

Task	**Layout**	**Items**	**Answer**	**Image size (px)**	**Images**
Shape counting	3–7 × 3–7	5–47 (19.0)	0–22 (3.2)	490–1130 × 580–1340	800
Angle ranking	1–3 × 3–9	3–23 (10.4)	Ranking	490–1450 × 170–490	600
Size ranking	1–3 × 5–7	4–13 (7.7)	Ranking	1280–1790 × 260–770	600
Clock ranking	1–3 × 3–6	3–18 (9.3)	Ranking	490–970 × 170–490	1,200
Transformation	1 + 4 panels	1 shape	Option (A–D)	660 × 1000	600

B.2 TASK GENERATION

We pre-define shapes from 11 classes: square, rectangle, trapezium, parallelogram, regular pentagon, equilateral triangle, isosceles triangle, right triangle, right isosceles triangle, circle, and ellipse.

Shape counting. We randomly sample shapes from the 11 classes to fill the cells. Some cells are left empty. We add marks as follows: squares and rectangles have right-angle marks at all

corners; squares, pentagons, and equilateral triangles have one tick on every side; rectangles and parallelograms have one tick on one pair of opposite sides and two ticks on the other; isosceles triangles have one tick on each equal side; right triangles have a right-angle mark; right isosceles triangles have a right-angle mark and one tick on each leg; circles have two dotted radii, each with one tick. Trapezia and ellipses have no marks. Each shape has a random orientation and a random size, and classes with free proportions (e.g., rectangles, isosceles triangles, ellipses) also get random proportions. The target class is drawn from all 11 classes, not only those present, so the answer can be zero. The gold count follows class inclusion: a square also counts as a rectangle, parallelogram, and trapezium; a rectangle also counts as a parallelogram and trapezium; a parallelogram also counts as a trapezium; equilateral and right isosceles triangles also count as isosceles triangles; a right isosceles triangle also counts as a right triangle; and a circle also counts as an ellipse. The question uses one of five templates, such as "How many parallelograms are there?" and "What is the total number of circles in the image?"

Angle ranking. Angles are whole degrees between 5° and 355°, and any two angles in a scene differ by at least 5°; unused cells are left empty. All angles in a scene has a ray that share the same direction (0°, 90°, 180°, or 270°) and the turning direction (clockwise or anticlockwise). The ranking direction is drawn at random for each scene.

Size ranking. All shapes in a scene belong to one of the 11 classes above, drawn without marks, and share the same rotation (90°, 180°, 270°, or 360°) and proportions (aspect ratio between 1.2 and 2.2 for classes with free proportions). Each shape has a scale factor between 0.10 and 0.95 that sets its linear size, and any two scale factors in a scene differ by at least 0.05. The ranking direction is drawn at random for each scene.

Clock ranking. Times are drawn at random from the 1,440 minutes of the day, and any two clocks in a scene differ by at least 5 minutes. The hour hand moves with the minutes (e.g., at 2:30 it points halfway between 2 and 3). The ranking direction is drawn at random for each scene.

Transformation prediction. Each image has five panels: the source panel (O) at the top and four options below it in a 2×2 layout. The source shape is one of the 11 classes, drawn without marks near the panel center, and P is placed near the center. Each chain has three steps. A step is one of: a rotation about P (90° clockwise or anticlockwise, 180°, or 270° clockwise); a dilation about P by 0.5 or 2; a reflection across a horizontal or vertical line XY at a random position, used at most once per chain; or a translation up, down, left, or right, used only as the last step. We discard chains that move the shape outside the panel. The point P and the line XY appear in all five panels. An example question is "(O) is rotated 90 degrees clockwise around P, then reflected across XY, then translated left. Which option shows the final result?" Wrong options are chosen at random from a pool of faulty chains: one step changed to a similar but wrong transformation (e.g., a rotation by a different angle or about the shape's center, a reflection across a different line, a dilation with a different scale or center, or a translation in a different direction), one step removed, one step applied alone, or the steps applied in reverse order. If fewer than three valid options remain, we add a random single step. We reject any option that looks nearly the same as another (mean pixel difference below 3 on a 0–255 scale). Because the question does not state how far a translation moves the shape, we also reject any option that shows the correct shape moved in the same direction.

B.3 Reference Conditions

Numbering. In shape counting, a label "(k)" is printed above each shape, numbered row by row across all shapes. In clock ranking, the numbers 1–12 are printed inside the dial at the hour ticks; the two incomplete versions print only (3, 6, 9, 12), or (6, 12).

Shading. Shading fills interiors with light gray instead of white; outlines are unchanged. It fills every shape in shape counting and size ranking, the region between the two rays indicated by the red arc in angle ranking, the whole dial face in clock ranking, and the shape in all five panels in transformation prediction.

Grid lines. In shape counting and the three ranking tasks, light gray lines are drawn every 15 px. In transformation prediction, lines are drawn every 30 px in each 300×300 px panel; a translation moves the shape by exactly two grid cells.

C Evaluation Details

C.1 Prompts

Each task uses a single prompt template, shown below. The placeholder {question} is replaced by the question for each sample, e.g., "How many parallelograms are there?" for shape counting or "Rank the clocks from earliest to latest by their labels." for clock ranking. The prompt depends only on the task and the question: within each paired set, all reference conditions receive identical text, and only the image differs.

Shape counting.

```
{question}

**Rules**:
- Rectangles include squares.
- Parallelograms include rectangles.
- Trapeziums include parallelograms.
- Isosceles triangles include equilateral triangles.
- Ellipses include circles.

**Instructions**:
You may think step-by-step and show your reasoning to carefully count
the shapes.
Once you have finished your reasoning, you MUST enclose your final
numerical integer count within <answer> tags.

Example format:
[Your reasoning and step-by-step counting here...]
<answer>7</answer>
```

Angle ranking.

```
{question}

**Instructions**:
You are tasked with ranking the angles indicated by the arcs based on
their visual size.
The angles are identified by numerical labels in parentheses, such as
(1), (2), (3), etc.

You may think step-by-step and show your reasoning to carefully compare
 the angles.
Once you have finished your reasoning, you MUST enclose your final
answer within <answer> tags as a standard array of integers
representing the labels. Do not include the parentheses in your final
array.

Example format:
[Your reasoning and step-by-step comparison here...]
<answer>[3, 1, 2, 4]</answer>
```

Size ranking.

```
{question}

**Instructions**:
```

```
You are tasked with ranking the shapes based on their visual size.
The shapes are identified by numerical labels in parentheses, such as
(1), (2), (3), etc.

You may think step-by-step and show your reasoning to carefully compare
 the sizes.
Once you have finished your reasoning, you MUST enclose your final
answer within <answer> tags as a standard array of integers
representing the labels. Do not include the parentheses in your final
array.

Example format:
[Your reasoning and step-by-step comparison here...]
<answer>[3, 1, 2, 4]</answer>
```

Clock ranking.

```
{question}

**Instructions**:
You are tasked with ranking the clocks based on the times shown.
The clocks are identified by numerical labels in parentheses, such as
(1), (2), (3), etc.
Use the AM/PM marker when comparing times.

You may think step-by-step and show your reasoning to carefully compare
 the times.
Once you have finished your reasoning, you MUST enclose your final
answer within <answer> tags as a standard array of integers
representing the labels. Do not include the parentheses in your final
array.

Example format:
[Your reasoning and step-by-step comparison here...]
<answer>[3, 1, 2, 4]</answer>
```

Transformation prediction.

```
{question}

**Instructions**:
You may think step-by-step and show your reasoning to carefully apply
the transformations in order.
Once you have finished your reasoning, you MUST enclose your final
answer within <answer> tags as a single capital letter.

Example format:
[Your reasoning and step-by-step transformation here...]
<answer>B</answer>
```

C.2 Reference-strategy annotation

We manually assign each shape-counting response of Gemini-3.1-Flash-Lite and GPT-5.4-mini on plain and numbered images (800 responses) to one of four categories: (A) numbered reference, which uses object numbers as addresses, such as "(3)" or "shape 7"; (B) row-and-column reference, which uses a systematic row, column, or cell scheme, such as "row 2, column 3"; (C) vague spatial reference, which uses only phrases such as "top left" without a stable scheme; and (D) unclear or no reference, which gives the answer directly or cannot be classified. We call categories A and B *systematic references*. We call categories C and D *ambiguous* references.

C.3 PAIRWISE ACCURACY

A ranking of n items contains $n(n-1)/2$ item pairs. For a valid predicted ranking, pairwise accuracy is the fraction of these pairs placed in the same order as in the gold ranking; it equals $(\tau + 1)/2$, where τ is Kendall's rank correlation. We average it over the scenes with a valid ranking. An answer is invalid if it cannot be parsed or does not list every item label exactly once.

C.4 WRONG-SIDE READINGS IN ANGLE RANKING

We use only angles below 90° or above 270°. For such an angle with true value θ, the wrong-side value $360^\circ - \theta$ lies on the other side of 180° and differs from θ by more than 180°, so the two readings put the angle in very different places in the ranking. Within the angle's scene, let r be its correct position in ascending order, that is, the number of other angles smaller than θ, and let r' be the position it would take if read as $360^\circ - \theta$, that is, the number of other angles smaller than $360^\circ - \theta$. Let $\hat{r}$ be its position in the predicted ranking, also in ascending order. We count a wrong-side reading if the angle is placed closer to r' than to r, that is, $|\hat{r} - r'| < |\hat{r} - r|$. We skip angles with $r = r'$, for which the two readings give the same position, and use only scenes whose rankings are valid for both the plain and the shaded image. Because the positions are counted within each scene, the rule needs no threshold and adapts to the number of angles in the scene. As a reference, we apply the same rule to 200 random rankings of each scene.

C.5 GRID-USE JUDGMENT

For transformation prediction, an LLM judge (Claude Sonnet 5) reads each response of Gemini-3.1-Flash-Lite and GPT-5.4-mini with grid lines (400 responses) and decides whether it uses the grid for measurement. A response does so if it gives at least one position, distance, or size in grid cells, units, or squares, or as grid coordinates, for the shape, its vertices, the point P, the line XY, or the shape in an option panel. Rough or tentative measurements count. Coordinates count only if they describe positions in the image, not symbolic rules such as "(x, y) becomes $(-y, x)$", and numbers that only restate the question, such as rotation angles or dilation factors, do not count. The judge sees the question and the response but not the image, and it quotes the part of the response that supports each positive decision; we checked that every such quote is supported by the response. We call a response that measures with the grid a systematic reference and any other response an ambiguous reference. The judge prompt is shown below; {shape_class}, {question}, and {response} are filled in for each response.

```
You are analyzing a model's response to a geometric transformation task
. The model saw
an image with a source panel (O) showing a {shape_class} and a marked
point P (and a
line XY if a reflection is involved), plus four option panels labeled A
 to D. All five
panels have evenly spaced light gray grid lines. The model was asked:
"{question}".
You see only the model's written response, not the image.

Decide whether the response uses the grid to measure. Choose exactly
one:

(A) Uses the grid for measurement: the response gives at least one
position, distance,
   or size in grid cells, units, or squares, or as grid coordinates,
   for the shape,
   its vertices, the point P, the line XY, or the shape in an option
   panel. Examples:
   "the rectangle is 2 units wide and 3 units tall", "P is at (0, 0)
   and the top
   vertex is at (1, 2)", "the shape moves 2 cells to the left", "in
   option B the
   shape is 4 squares from P".
```

```
(B) Does not use the grid for measurement: the response does not
mention the grid, or
   mentions it only in general terms without giving any number of cells
    or grid
   coordinates. Examples: "the shape moves left on the grid", "P stays
   in the same
   place on the grid", "option C matches the expected position".

Rules:
1. Rough or tentative measurements count as (A).
2. Coordinates count as (A) only if they describe positions in the
image, such as
  positions read from the grid or measured relative to P. Symbolic
  rules applied to
  unspecified points, such as "(x, y) becomes (-y, x)", do not count by
   themselves,
  and neither does only setting P as the origin.
3. Numbers that only restate the question, such as rotation angles,
dilation factors,
  or the number of steps, do not count.
4. Only grid cells, units, squares, or grid coordinates count; pixels
and vague words
  such as "slightly" do not.
5. A measurement counts even if it appears in a heading or on a
different line from
  the shape's name.

Return {"grid_use": "A" or "B", "evidence": "a short exact quote
supporting your
choice, or an empty string if there is none"}

Response:
{response}
```

D OTHER RESULTS

Table 5 lists pairwise accuracy for angle and size ranking, Table 6 gives the wrong-side analysis of angle ranking. Table 1 splits size-ranking pairwise accuracy by size gap.

Table 5: Pairwise accuracy (%) on angle and size ranking, averaged over valid rankings.

	ANGLE RANKING		SIZE RANKING		
MODEL	Plain	Shading	Plain	Shading	Grid lines
Claude-Opus-4.7	88.5	90.9	96.8	97.4	97.6
Gemini-3.1-Flash-Lite	76.4	85.2	95.3	96.2	95.8
GPT-5.4-mini	80.1	80.5	96.2	96.7	96.6
Qwen3-VL-30B-A3B	59.7	68.0	92.4	93.2	93.1
InternVL3.5-38B	69.2	76.1	91.1	93.6	91.4

Table 6: Wrong-side readings in angle ranking: the share (%) of angles below 90° or above 270° that are placed closer to their wrong-side position than to their true position (Appendix C). Scenes are those with valid rankings for both the plain and the shaded image. Random is the rate for random rankings of the same scenes.

			Wrong-side rate (%)		
Model	**Scenes**	**Angles**	Plain	Shading	Random
Claude-Opus-4.7	200	1,030	3.3	1.7	47.8
Gemini-3.1-Flash-Lite	186	925	18.4	8.5	47.8
GPT-5.4-mini	181	914	13.3	13.8	47.8
Qwen3-VL-30B-A3B	95	377	31.6	27.9	47.0
InternVL3.5-38B	161	716	23.9	13.1	47.7