arXiv:2610.38957v1 [cs.CV] 28 Oct 2026

What do MLLMs know about "Waldo"? Evaluating Visual Reasoning under Multi-Instance Multi-Object Challenges

Saumya Mundra *saumyamundra@gmail.com*
IIIT Hyderabad

Ajoy Mondal *ajoy.mondal@iiit.ac.in*
IIIT Hyderabad

Vineeth N Balasubramanian *vineethnb@iith.ac.in*
Microsoft Research India
IIT Hyderabad

C. V. Jawahar *jawahar@iiit.ac.in*
IIIT Hyderabad

Abstract

Recent advances in Multimodal Large Language Models (MLLMs) have led to significant progress in visual reasoning capabilities. However, existing benchmarks curate image-question pairs pertaining to object categories with few instances, enabling trivial object referencing and limited visual ambiguity. In this work, we introduce **MIMO-Bench**, a benchmark for evaluating MLLMs in a **M**ulti-**I**nstance per **M**ulti-**O**bject categories (MIMO) setting. Each image-question pair in our benchmark pertains to instances belonging to object categories with multiple instances. The presence of such similar feature distractors causes hindrance in basic visual reasoning tasks, necessitating the referring expression to be composed of multiple visual subtasks to uniquely and unambiguously identify the target object. MIMO-Bench comprises 3,802 image-question pairs across attribute binding and numerical counting tasks. Despite impressive performances on existing benchmarks, both SOTA open-source (InternVL, QwenVL) and proprietary (Gemini, GPT-5) models exhibit substantially lower performance on MIMO-Bench, highlighting their limitations in the MIMO setting. The benchmark will be made publicly available.

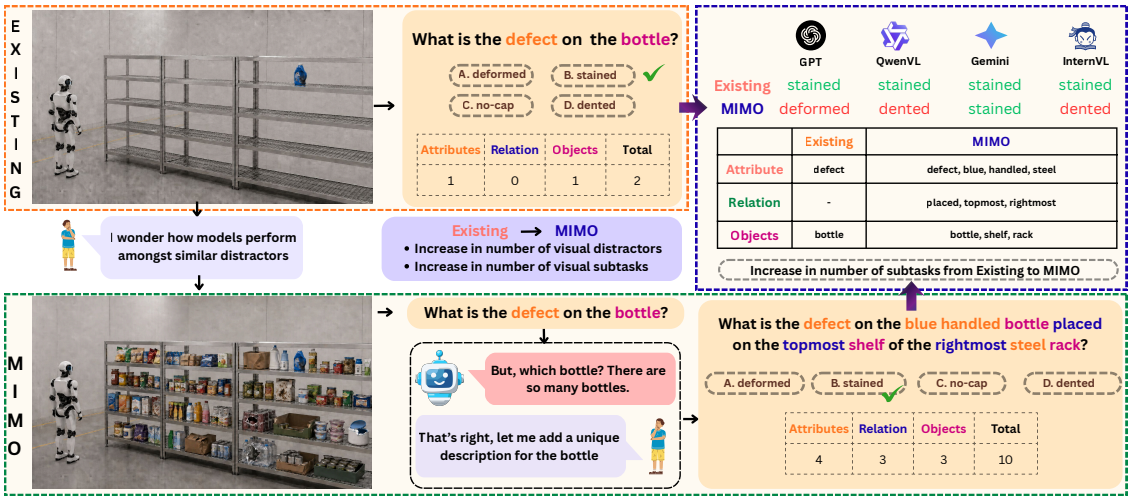
1 Introduction

Answering a rudimentary question such as *"What is the defect on the bottle?"* is a trivial task for humans when the queried image contains one bottle. However, asking the same question upon introducing multiple bottles in the image immediately raises ambiguity, leading humans to naturally ask "Which bottle?". One way to counter such ambiguity is to ensure that our reference to the bottle is as detailed as possible, leading us to refine our question to *"What is the defect on the blue handled bottle placed on the topmost shelf of the rightmost steel rack?"*. Humans can still answer accurately by using pre-attentive visual features (Wolfe et al., 1989) and sequentially integrating information from the question and the visual scene (Treisman & Gelade, 1980). However, performing this task under the same time constraints often leads to illusory conjunction such as misbinding attributes like color or position.

Multimodal Large Language Models (MLLMs) (Liu et al., 2023b; Wang et al., 2024; Zhu et al., 2025; Alayrac et al., 2022) have unified vision and language understanding, enabling remarkable progress across diverse tasks, including text-centric (Mathew et al., 2021; 2022), STEM-based (Lu et al., 2024; He et al., 2024), and real-world vision-centric reasoning (Tong et al., 2024; Chen et al., 2024a; Fu et al., 2024; Liu et al., 2024; Li et al., 2024). Vision-centric reasoning, in particular, depends heavily on accurate visual grounding; the ability to locate and reason about specific objects in complex visual scenes.

Benchmarks such as MMStar (Chen et al., 2024a), BLINK (Fu et al., 2024), CV-Bench (Tong et al., 2024), and NaturalBench (Li et al., 2024) have advanced the evaluation of vision-centric reasoning, posing tasks such as *numerical counting* (e.g., "Count the number of table lamps in the image?"), *spatial reasoning* (e.g., "Is the person to the left of the pole?"), and *attribute recognition* (e.g., "What is the primary color of the car?"). However, image-question pairs in most existing benchmarks pertain to object categories containing very few instances; either because the curated images contain only a few (often only one) instances per object category, or because the questions are targeted at objects belonging to single-instance categories, despite multi-instance object categories being present. As a result, target objects are easily distinguishable, and simple referring expressions such as `<the_object_category>` (e.g., "the man" or "the car") suffice. While this design facilitates a controlled isolated evaluation of reasoning ability, it limits the assessment of fine-grained reasoning. As modern MLLMs such as InternVL3 (Zhu et al., 2025), Qwen3-VL (Bai et al., 2025a), and GPT-5 (OpenAI, 2025b) perform basic vision tasks well, a natural question arises: how do MLLMs perform in settings featuring **M**ulti-**I**nstance per **M**ulti-**O**bject categories (**MIMO**)?

Figure 1: **Existing vs. MIMO setting.** MIMO introduces dense multi-instance scenes with more visual distractors and richer referring expressions.



The MIMO setting is characterized by a target instance co-occurring with other instances of the same object category that act as similar-feature distractors. Presence of such distractors hurt primary visual reasoning tasks such as attribute binding and numerical counting (Campbell et al., 2024; Izadi et al., 2025). This reflects the classical binding problem (Roskies, 1999) in visual cognition and suggests that a single visual reasoning task in MIMO is harder. To add to this, the MIMO setting necessitates that the referring expression be composed of multiple visual subtasks. For example, we can not simply refer to the bottle in the MIMO-image in Fig. 1 as "*the bottle*". Rather, we combine *attributes/actions* and *relations* of neighboring *objects* to refer to the bottle uniquely as "the *blue handled bottle placed on* the *top-most shelf of* the *right-most steel rack*". Such referring expressions impose multiple visual subtasks (Johnson et al., 2017), including object recognition, attribute/action recognition, and relational reasoning. As a result, both the number of visual subtasks and the difficulty of each subtask amplifies in MIMO (Fig. 1).

To evaluate MLLMs under the MIMO setting, we introduce **MIMO-Bench**, a benchmark comprising 3,802 image-question pairs for attribute binding and numerical counting. MIMO-Bench consists of two primary sources of difficulty: (i) *presence of similar distractors* induced by the existence of multiple instances per multiple object categories and (ii) *complex referring expressions* that pose multiple visual subtasks. Standard accuracy (Tong et al., 2024; Chen et al., 2024a) captures neither whether a model consistently binds all of an instance's attributes nor whether its counting errors are global or localized. Thus, we define **Group Accuracy** for attribute binding and, for numerical counting, complement **Mean Relative Accuracy (MRA)** (Yang et al., 2025) with **Global MRA** and **Local MRA**. To isolate the role of referring expressions, we replace them with **visual prompts** that mark the target directly, and we further show that performance degrades as scene distractors increase. Our contributions and findings are as follows:

1. We introduce MIMO-Bench, a benchmark of 3,402 attribute-binding and 400 numerical-counting questions for evaluating MLLMs in dense Multi-Instance per Multi-Object scenarios.

2. Current MLLMs struggle substantially on MIMO-Bench, with state-of-the-art zero-shot performance on real images reaching only about 52.8% $\mathcal{I}_{\mathrm{ACC}}$ on attribute binding and 56.7% $\mathcal{I}_{\mathrm{MRA}}$ on numerical counting.

3. Referring expressions and similar-feature distractors are key bottlenecks, as visual prompting improves $\mathcal{I}_{\mathrm{ACC}}$ by approximately 15.5 points on average, while our graying and inpainting experiments show that reducing visual distractors further improves performance.

4. Iterative visual access is a promising direction, as naive relevant-region extraction consistently improves performance across models, while Chain-of-Thought prompting does not reliably help.

2 Related Work

Vision-centric multimodal benchmarks. Recent MLLM families such as LLaVA (Liu et al., 2023b), InternVL (Chen et al., 2024b), QwenVL (Bai et al., 2023), GPT-5 (OpenAI, 2025b), and Gemini-2.5-Pro (Comanici et al., 2025) perform strongly on general vision–language tasks, as tracked by vision-centric benchmarks where answers must be inferred from the image rather than from dataset or language priors (e.g., MMStar (Chen et al., 2024a), BLINK (Fu et al., 2024), RealWorldQA (xAI, 2024), CV-Bench (Tong et al., 2024), NaturalBench (Li et al., 2024)). Others isolate specific difficulties: V*Bench (Wu & Xie, 2024) targets visual search for small objects, while HR-Bench (Wang et al., 2025b) and MME-RealWorld (Zhang et al., 2025) stress high-resolution perception. MIMO-Bench adds a new axis: dense same-category scenes that raise distractor count and force referring expressions to take on richer compositions to disambiguate the target.

Referring expressions. A referring expression is classically studied in natural language generation as a *distinguishing description*: a phrase singling out an intended referent from competing distractors while respecting the Gricean maxims (Grice, 1975; Dale & Reiter, 1995). Disambiguation against similar candidates is thus intrinsic to its purpose, while its form ranges from a bare category label to a long relational description. Vision–language work brought referring expressions into real-image grounding through ReferItGame (Kazemzadeh et al., 2014) and the MS-COCO RefCOCO/RefCOCO+/RefCOCOg suite (Yu et al., 2016; Mao et al., 2016). Referring Expression Comprehension (Yu et al., 2016; Chen et al., 2025) and related tasks, including phrase grounding (Plummer et al., 2015), referring segmentation (Hu et al., 2016; Liu et al., 2023a), Described Object Detection (Xie et al., 2023), and Referring Expression Instance Retrieval (Hao et al., 2025), treat reference resolution or localization as the final output. Closest to our setting, Ref-Adv (Dong et al., 2026) also pairs compositional expressions with same-category distractors, but averages 4.01 distractors per case and evaluates localization itself. MIMO-Bench instead places reference resolution *upstream* of attribute binding and hierarchical counting, in scenes averaging 47.7 same-category distractors per target,[1] so that resolving the reference is a prerequisite rather than the task. To construct expressions that disambiguate a target among these distractors, we adopt the attribute–object–relation framing used to characterize fine-grained vision–language structure (Yuksekgonul et al., 2023; Ma et al., 2023; Hsieh et al., 2023), composing attributes, actions, and inter-object relations into expressions that uniquely identify the referent.

Cognitive foundations. MIMO-Bench's difficulties are rooted in human visual cognition. Locating a target among similar items is visual search: without pop-out, it is slow and serial (Treisman & Gelade, 1980; Wolfe et al., 1989). A candidate's features must then be correctly integrated (attribute binding), yet under clutter humans misbind color, material, or shape, producing illusory conjunctions (Treisman & Schmidt, 1982; Roskies, 1999). Enumeration is similarly constrained: small sets are subitized, but counting degrades as items grow and feature variability drops (Kaufman et al., 1949). VLMs inherit these failures: multi-object performance depends on the probability of feature interference among co-occurring objects (Campbell et al.,

[1] Not a like-for-like measure: Ref-Adv's 4.01 counts its authors' curated hard distractors (Dong et al., 2026), whereas ours counts every same-category instance SAM3 (Carion et al., 2026) detects per target prompt, including non-confusable ones and detector errors. Read it as a difference in scene density, not distractor difficulty.

2024), and counting worsens with more instances and less diversity (Ranjan et al., 2021; Paiss et al., 2023; Acharya et al., 2019). MIMO-Bench brings search, binding, and enumeration into dense natural scenes where same-category instances act as similar-feature distractors.

Internal grounding mechanisms. A complementary line of work studies how MLLMs internally allocate attention to visual evidence and how grounding failures emerge. Layer-wise analyses show attention progressively concentrating on task-relevant regions, while cluttered scenes can fragment attention and cause smaller targets to be under-attended (Lyu et al., 2025; Liu et al., 2026; Lyu et al., 2026b). Other work localizes grounding failures to internal image-specific features and improves grounding through representation steering (Lyu et al., 2026a). These approaches intervene inside the model; our interventions are behavioural, modifying the visual input through marking, graying, or cropping and measuring the resulting change in MIMO performance without identifying the internal mechanism responsible.

Table 1: **Positioning MIMO-Bench.** Each row places a benchmark group along axes characterizing its construction: image source (real/synthetic), template-based question generation, compositional referring expressions (REs), image resolution (Hi/Lo), and if same-category distractor density is a controlled design axis (✓), incidental (∼), or not considered (✗). # subtasks is the mean number of visual subtasks per query. Mean Distr. is the mean number of same-category instances per target, excluding the target itself; unless noted otherwise, values are computed using our SAM3-based protocol. †For Ref-Adv, we use the average same-category distractor count reported by its authors.

Lineage	Benchmark(s)	Real/Syn	Template	Comp. REs	Image Res.	Distractor	Mean Distr.	# subtasks
Compositional	CLEVR (Johnson et al., 2017)	✗/✓	✓	✓	✗/✓	✗	1.74	7.2
	GQA (Hudson & Manning, 2019)	✓/✗	✓	✓	✗/✓	✗	1.67	3.1
Vision-centric (general)	MMStar, NaturalBench, VQAv2 (Goyal et al., 2017)	✓/✗	✗	✗	✗/✓	✗	2.49/2.11/1.46	3.9/2.7/2.6
Visual difficulty	V*Bench	✓/✗	✗	✗	✓/✗	∼	0.12	2.4
	HR-Bench, MME-RealWorld	✓/✗	✗	✗	✓/✗	∼	2.01/6.54	4.7/3.9
Referring expression	Ref-Adv (Dong et al., 2026)	✓/✗	✗	✓	✗/✓	✓	4.01†	6.6
Ours	**MIMO-Bench**	✓/✓	✓	✓	✓/✓	✓	**47.7**	**9.6**

3 MIMO-Bench

3.1 Overview

We introduce MIMO-Bench, a benchmark to evaluate reasoning capabilities of MLLMs in scenes with multiple instances of the same category. These *dense same-category instances* act as natural visual distractors and necessitate *compositional referring expressions (REs)* to disambiguate them, raising their subtask count. MIMO-Bench spans two reasoning tasks, *attribute binding* and *numerical counting*, constructed from real and synthetic images. Table 1 positions MIMO-Bench amongst existing benchmarks, showing MIMO-Bench treats distractor-driven ambiguity as a controlled design axis and has a high mean subtask count of 9.6.

3.2 Tasks

We instantiate the MIMO setting through **attribute binding** and **numerical counting**, as defined below.

Attribute Binding (AB) is posed as a multiple-choice task. It evaluates a model's ability to correctly associate a target object with its specified attributes and the attributes of its structured components, where the task requires attaching *each* specified attribute to the *same* target instance[2] (see Fig. 3). In cognitive science, *binding* refers to integrating multiple perceptual features such as color, shape, and material into a coherent object representation. Both humans and MLLMs are susceptible to the *binding problem*, where the presence of visually similar distractors leads to incorrect attribution of attributes across nearby objects, motivating attribute binding as a task for our benchmark.

[2] $\mathcal{G}_{\text{ACC}}$ (Eq. 1) scores each target group by the fraction of its valid attribute questions answered correctly and weights every group equally; it is not an all-attributes-correct criterion. The qualifier *group* denotes the unit of aggregation, not the strictness of the criterion.

Numerical Counting (NC) is posed as an open-ended counting task, where the model predicts the number of target instances matching a specified RE. Each count is defined over the region delimited by its RE, and counts are organized hierarchically so that region-bounded subgroup counts must sum to their parent group, a consistency check enforced during annotation. Prior studies (Campbell et al., 2024) show that low visual diversity among objects reduces counting accuracy, and human perception itself struggles when the number of objects exceeds four (Kaufman et al., 1949). By combining high feature similarity and object counts, MIMO-Bench evaluates how MLLMs perform numerical counting under multi-instance ambiguity.

Referring Expressions (REs): Necessity. Both tasks are *referential*: solving either requires comprehending an RE to localize the target(s) and performing the task-specific operations. A category-level reference, e.g., "the plant", is insufficient and ambiguous in MIMO scenes. Each target must be specified by an RE that is *descriptive*, *informative* and *unambiguous* (Grice, 1975). Long expressions often become necessary and natural owing to the dense scene structure. In attribute binding, the RE identifies a single target instance, whereas in numerical counting, it specifies the group of instances to be counted.

REs: Structure. An RE may combine a target's intrinsic attributes, e.g., color, material, or size; actions or states, e.g., walking or sitting; and relations with neighboring objects. For example, *"the plant whose pot is vertically striped with yellow and white"* or *"the woman second closest to the person second closest to the woman near the trolley with the pink poster"*. Counting targets are specified analogously, e.g., *vertically striped pillows*, *plants on the topmost shelf*, or *transparent utensils with purple lids placed between the gas stove and the blue kettle*. The difficulty of REs arises from their composition, a model must comprehend the attributes, actions or states, and inter-object relations it contains, and ground them to the correct object or group of objects in the scene. These subtasks correspond to components of a visual scene graph (Johnson et al., 2017; Ma et al., 2023); they are often trivial in single-object settings but become non-trivial under MIMO conditions as REs in MIMO-Bench require solving approximately 9.6 such visual subtasks on average, twice as many as in simple references in existing vision-centric benchmarks, as shown in Tab. 1. Fig. 2 summarizes **RE length and composition** in terms of relations, attributes/actions, objects and their combinations.

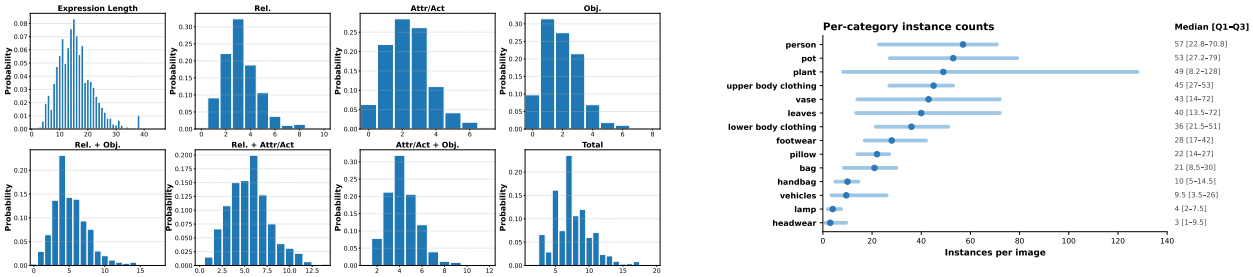


Figure 2: **Dataset statistics for MIMO-Bench.** Distributions of RE length and composition (left), and per-category instance counts over images containing each category (right).

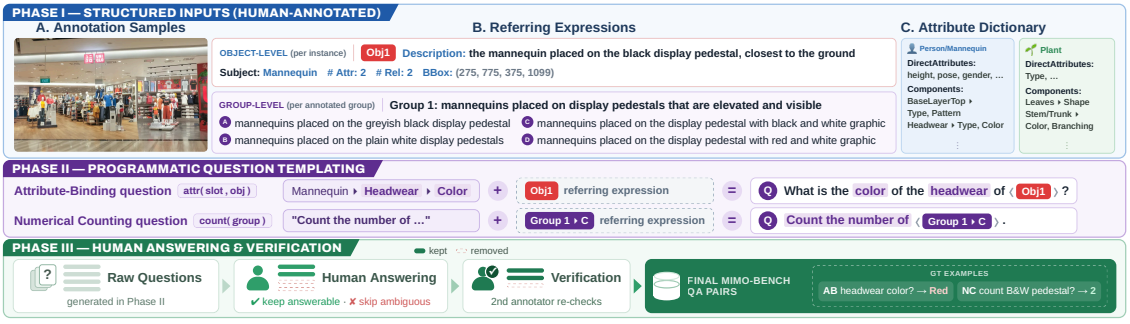


Figure 3: **MIMO-Bench Creation Pipeline**: **AB** questions are generated by mapping instance annotations to the attribute dictionary, producing raw questions. **NC** questions are generated from REs describing groups. Human answering and verification retain only visually grounded and unambiguous QA pairs.

3.3 Benchmark Construction

We develop a semi-automatic benchmark construction pipeline to ensure high-quality, linguistically precise, and visually grounded QA pairs, as shown in Fig. 3.

Data Collection: Our benchmark requires scenes containing multiple instances per object category. We sample real images from SA-1B (Kirillov et al., 2023), selecting images based on the number of annotated bounding boxes and scene depth. As SA-1B is predominantly outdoors, we supplement it with synthetic images to broaden coverage of indoor scenes, object categories, and attribute configurations that are hard to obtain at scale from real-image sources. Synthetic images are generated using Gemini-2.5-Pro (Comanici et al., 2025) with carefully designed prompts to emulate MIMO conditions. We also experimented with Grok-4 and Flux for image generation; however, these models produced lower realism, feature diversity, and prompt adherence compared to Gemini-2.5-Pro. A pilot across several open and closed models on Gemini-generated images showed no self-preference advantage for Gemini-2.5-Pro, with a non-Gemini model scoring highest. Data sampling, generation protocol and qualitative comparison, and self preference-bias are detailed in the appendix. Fig. 2 reports per-category instance-count distributions over images containing that category. Categories such as person, vase, and pot contain several tens of instances per image on average, showing that targets are typically surrounded by many same-category distractors. Distractor count was calculated using SAM3 (Carion et al., 2026), with protocol detailed in the appendix.

Data Annotation: For each image, objects are selectively annotated using polygonal bounding masks[3], with each object described by one or more REs following Sec. 3.2. We take two precautions to keep these expressions operationally unambiguous and comprehensible, treating an RE as valid only when annotators can identify its intended target without disagreement and flagged cases are revised or removed. First, we use models only as a lightweight linguistic sanity check. We prompt several models to state what they understand each RE to mean, on the hypothesis that an expression which is not overly convoluted will be interpreted consistently across models. Models agreed on RE interpretation 85% of the time, indicating the expressions are interpreted consistently rather than being linguistically convoluted. Second, annotators who answer and verify questions flag any expression they find ambiguous or in disagreement with its referenced object. At the answering stage, 6.91% of all REs were flagged; these were re-annotated to remove ambiguity or simplify the language and then returned to the verification round together with the previously-agreed expressions, after which the overall disagreement rate across the full set dropped to 2%, with the remaining cases discarded. The complete annotation protocol is detailed in the appendix material.

Question–Answer Generation: From the prior stage, we obtain images with annotated objects, where each object is annotated with polygon masks and one or more REs. For each annotated object category, GPT-5 (OpenAI, 2025b) is used as an auxiliary tool to propose candidate attribute keys for a hierarchical attribute dictionary describing both object-level attributes and attributes of its components, as shown in Fig. 3. The dictionary is manually refined to ensure broad coverage and mutual exclusivity among attributes.

Each object–RE pair is matched to the corresponding object category key in the attribute dictionary, and this match is passed through a template-based question generator, generating raw unanswered multiple-choice questions pertaining to dictionary attributes. The options are hard negatives to avoid ambiguity. For numerical counting, we append *"Count the number of"* to annotated sets to generate questions. These generated unanswered questions are passed through an iterative human answering and verification pipeline. In the first answering round, only those candidate questions are answered whose queried attribute or object is physically present, clearly visible, and unambiguous in the image, and candidate questions over absent or ambiguous attributes are discarded before verification. Annotators are instructed to discard only on these grounds rather than arbitrarily, so that skipping acts as the principal filter from the exhaustive raw pool to visually grounded QA pairs. The criterion is whether an answer is visually recoverable, not whether the question is easy: candidates were never discarded for being difficult or for being answered incorrectly, so dense scenes and complex REs remain in the benchmark. Thus, the human baseline simply measures human capability on MIMO-Bench, rather than reflecting a benchmark made easier through difficulty-based filtering. All remaining QA pairs undergo an additional round of human verification to ensure correctness

[3]We release these target masks and the corresponding bounding-box coordinates with MIMO-Bench, so that alternative grounding and localization interfaces can be evaluated without reconstructing the annotations.

and minimize ambiguity, resulting in a high-quality benchmark. We discuss the human answering in more detail in the appendix, along with MIMO-Bench scalability.

MIMO-Bench contains *3,402 attribute-binding* questions and *400 numerical-counting* questions. Each attribute-binding QA pair corresponds to an image–object–RE–attribute tuple, and each numerical-counting QA pair to an image–target-group–RE tuple.

4 Evaluation on MIMO-Bench

4.1 Experiment Setup

We evaluate API-accessed models Gemini-3.5-Flash (Google DeepMind, 2026b), Gemini-3.1-Pro-Preview (Google DeepMind, 2026a), Gemini-2.5-Pro/Flash (Comanici et al., 2025), GPT-5 (OpenAI, 2025b), GLM-4.5V (GLM-V Team et al., 2025), IVL3.5-241B-A28B (Wang et al., 2025a), and IVL3-78B (Zhu et al., 2025), and locally evaluated open-weight models Qwen3-VL (Q3) (Bai et al., 2025a), Qwen2.5-VL (Q2.5) (Bai et al., 2025b), the remaining InternVL3/3.5 (IVL3/IVL3.5) variants (Zhu et al., 2025; Wang et al., 2025a), and GLM-4.1V-9B-Thinking (GLM-V Team et al., 2025). We append size and variant to family abbreviations, e.g. Q3-32B-Instruct, IVL3.5-8B, G3.5-Flash. All experiments use the same task prompt and answer-format instruction across models. Model-specific components are limited to official HuggingFace/API chat templates, image preprocessing routines, and default model-card settings, unless specified otherwise. We did not observe answer-format violations; transient API failures were rerun. We also report 95% bootstrap confidence intervals (20,000 replicates), with the resampling protocol and analyses across different granularity levels detailed in Appendix C.9.

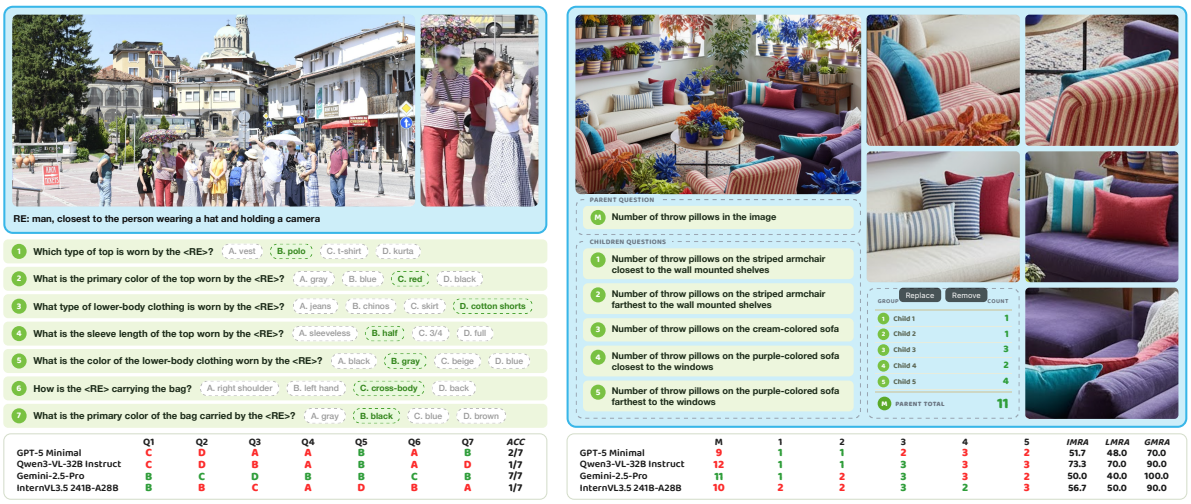


Figure 4: **AB** (left) and **NC** (right) tasks centered on a target instance or target group within an image.

4.2 Evaluation Metrics

Attribute Binding. We adopt Instance Accuracy ($\mathcal{I}_{\text{ACC}}$) as one of the evaluation metrics for AB. We refer to it as instance accuracy because each attribute-binding question corresponds to a single image–object–referring-expression–attribute quadruplet, which we treat as one instance. $\mathcal{I}_{\text{ACC}}$ is defined in Eq. 1, where N_A denotes the number of correct answers and N_Q represents the total number of questions.

To evaluate whether MLLMs form a *unified* and *grounded* representation of an object, we extend the evaluation beyond individual attributes. For a given instance in an image, we generate a *set* of AB image-question pairs pertaining to attributes that are visually *present*, *objective*, and define the object *uniquely* in the scene. The binding problem posits that models may fail to associate multiple attributes with a single instance. We introduce a Group Accuracy ($\mathcal{G}_{\text{ACC}}$) metric, similar to (Li et al., 2024), which aggregates per-attribute correctness within each target group while weighting target groups equally. When an instance is characterized

by only a few objective attributes, each attribute becomes crucial for identification as even a single misbinding would significantly impair the model's ability to correctly recognize the target object, as compared to objects with multiple identifiable attributes. Group Accuracy imposes a stronger penalty in the case when the objects are defined by few attributes, on account of groups being equally weighted. Formally:

$$\mathcal{I}_{\mathrm{ACC}} = \frac{N_A}{N_Q}, \quad \mathcal{G}_{\mathrm{ACC}} = \frac{1}{N_H}\sum_{h=1}^{N_H} \mathcal{H}_{\mathrm{ACC}}^{(h)}, \quad \mathcal{H}_{\mathrm{ACC}}^{(h)} = \frac{N_{HA}^{(h)}}{N_{HQ}^{(h)}}, \tag{1}$$

where $\mathcal{H}_{\mathrm{ACC}}^{(h)}$ is the accuracy of an individual group, N_H denotes the total number of groups in the benchmark, $N_{HA}^{(h)}$ and $N_{HQ}^{(h)}$ represent the number of correct answers and total questions within group h, respectively. Additionally, for completeness we also report a strict Exact Group Accuracy, which counts a target as correct only when all of its valid attribute questions are answered correctly, in Appendix C.2.

Numerical Counting performance is evaluated using the Mean Relative Accuracy ($\mathcal{MRA}$) metric (Yang et al., 2025). Given a prediction $\hat{y}$, ground truth y, and a confidence threshold θ, a prediction is considered correct if its relative error $\frac{|\hat{y}-y|}{y}$ is smaller than $1-\theta$. Formally,

$$\mathcal{MRA}(\hat{y}, y) = \frac{1}{|C|}\sum_{\theta\in C} \mathbb{1}\left(\frac{|\hat{y}-y|}{y} < 1-\theta\right), \quad C=\{0.5,0.55,\ldots,0.95\}, \quad \mathcal{MRA}(\hat{y},0)=\mathbb{1}(\hat{y}=0) \tag{2}$$

This convention treats zero-count questions as exact-match cases, preventing empty target sets from being discarded while avoiding division by zero. $\mathcal{MRA}$ provides an instance-level assessment of counting accuracy.

We utilize two simple additional measures to understand more about a model's NC capability in the MIMO setting. To do so, we structure the NC questions hierarchically by decomposing counting queries into spatially localized sub-questions. Consider the query: *"Count the number of throw pillows in the image."*. We decompose this parent query into a set of child queries, each corresponding to a distinct subregion where the target instances may occur, such as *"Count the number of throw pillows on the purple-colored sofa closest to the windows."* and *"Count the number of throw pillows on the striped armchair farthest to the wall mounted shelves."*, as shown in Fig. 4. At the annotation level, the global count satisfies $N = \sum_k N_k$, where N denotes the count over the full image and N_k denotes the count in the k-th localized subregion.

Using this structure, we measure an Individual Mean Relative Accuracy ($\mathcal{I}_{\mathrm{MRA}}$), a Global Mean Relative Accuracy ($\mathcal{G}_{\mathrm{MRA}}$), and a Local Mean Relative Accuracy ($\mathcal{L}_{\mathrm{MRA}}$). $\mathcal{I}_{\mathrm{MRA}}$ averages $\mathcal{MRA}$ over all numerical-counting queries, $\mathcal{G}_{\mathrm{MRA}}$ averages $\mathcal{MRA}$ over parent/global queries, and $\mathcal{L}_{\mathrm{MRA}}$ averages $\mathcal{MRA}$ over localized child queries under each parent. They are defined as

$$\begin{aligned} \mathcal{I}_{\mathrm{MRA}} &= \frac{1}{|\mathcal{Q}_{\mathrm{NC}}|}\sum_{q\in\mathcal{Q}_{\mathrm{NC}}} \mathcal{MRA}(q), \\ \mathcal{G}_{\mathrm{MRA}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \mathcal{MRA}(q_p) \qquad & \mathcal{L}_{\mathrm{MRA}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \left(\frac{1}{|\mathcal{C}(p)|}\sum_{c\in\mathcal{C}(p)} \mathcal{MRA}(q_c)\right). \end{aligned} \tag{3}$$

where $\mathcal{Q}_{\mathrm{NC}}$ denotes all numerical-counting queries, $\mathcal{P}$ denotes the set of parent queries, q_p denotes the parent query corresponding to parent p, and $\mathcal{C}(p)$ denotes the set of localized child queries associated with p. Importantly, $\mathcal{L}_{\mathrm{MRA}}$ does not evaluate whether a model's predicted child counts sum to its predicted parent count; instead, it isolates accuracy on localized child queries. Comparing $\mathcal{G}_{\mathrm{MRA}}$ and $\mathcal{L}_{\mathrm{MRA}}$ therefore reveals whether a model performs better when counting over the full scene or subregion respectively.

4.3 Result Analysis

Baselines and the human–model gap. We anchor all model scores against three references in Table 2: a random-chance baseline (mean $1/N$ over answer options), a majority-guess baseline (most common answer per question type), and a human baseline[4]. Humans operate near ceiling on both tasks and both splits, suggesting that the benchmark is far from saturated. The gap to the best model is large everywhere: on real AB the strongest model (52.8) trails humans by roughly 36 points, on real NC the strongest one-shot

[4]Performed by six humans who did not annotate MIMO-Bench, each solving 25 groups for AB and 15 for NC

Table 2: Evaluation results on MIMO-Bench, with AB and NC reported separately for real and synthetic splits; scores are percentages. **Baselines**: random chance is the mean $1/N$ over answer options, and majority guess picks the most common answer per attribute type. "Low", "medium" & "minimal" are reasoning efforts for the respective models. Abbreviations: Syn. = synthetic; – = NA

Method	Variant	Attribute Binding (AB)				Numerical Counting (NC)					
		Real		Syn.		Real			Syn.		
		$\mathcal{I}_{\text{ACC}}$	$\mathcal{G}_{\text{ACC}}$	$\mathcal{I}_{\text{ACC}}$	$\mathcal{G}_{\text{ACC}}$	$\mathcal{I}_{\text{MRA}}$	$\mathcal{L}_{\text{MRA}}$	$\mathcal{G}_{\text{MRA}}$	$\mathcal{I}_{\text{MRA}}$	$\mathcal{L}_{\text{MRA}}$	$\mathcal{G}_{\text{MRA}}$
Chance & Human Baselines											
Chance(Random)		15.8	15.6	15.8	15.3			–			
Chance(Majority)		34.3	35.5	34.3	33.1			–			
Human		89.3	86.7	85.7	87.5	87.7	86.4	90.1	89.6	87.4	92.0
API-accessed MLLMs											
Gemini-3.x	3.5 Flash Low	45.5	44.8	**73.0**	**75.6**	52.5	49.4	46.7	**66.9**	**66.0**	**67.2**
	3.1 Pro Preview Low	46.5	46.3	71.8	73.5	**56.7**	**55.8**	63.3	64.0	62.6	64.1
Gemini-2.5	Flash	51.3	51.9	48.1	42.1	30.0	28.3	38.3	47.2	41.2	51.4
	Pro	49.1	49.5	47.2	44.9	37.3	24.2	**71.7**	56.1	53.2	63.9
GPT-5	minimal	32.0	31.2	28.0	24.1	45.8	47.5	36.7	43.0	40.8	47.1
	medium	**52.8**	**53.4**	49.8	48.1	51.5	55.0	56.7	47.5	42.8	54.3
GLM	4.5V	50.3	50.0	41.7	33.8	33.9	36.2	38.3	47.3	48.5	38.6
InternVL3.5	241B-A28B	47.6	45.2	43.3	42.9	47.8	50.0	39.5	51.2	50.6	50.1
InternVL3	78B	40.2	40.3	37.6	35.0	38.9	43.3	38.3	36.6	34.2	34.3
Locally evaluated open-weight MLLMs											
Qwen3-VL	32B-Instruct	47.7	48.5	47.0	45.0	38.5	35.4	55.0	53.2	54.8	49.8
	8B-Instruct	42.4	42.0	36.2	32.5	34.2	28.3	43.3	47.6	48.9	35.5
	30B-A3B-Instruct	36.6	33.2	28.2	27.1	25.0	35.4	20.0	41.0	44.5	38.8
Qwen2.5-VL	72B-Instruct	38.6	38.7	50.6	45.1	30.8	35.8	23.3	46.2	47.0	43.1
	32B-Instruct	38.4	36.9	41.2	41.0	28.9	28.3	28.3	42.4	40.0	47.2
	7B-Instruct	35.5	35.2	36.4	34.3	27.3	30.0	31.7	33.9	33.5	27.2
InternVL3.5	38B	40.9	40.3	46.5	44.6	27.3	35.4	20.0	42.5	44.3	42.0
	14B	32.1	31.7	39.0	40.1	31.1	37.5	16.7	42.5	38.9	47.5
	8B	38.6	38.1	44.2	43.7	36.1	45.8	13.3	44.2	41.1	45.1
InternVL3	38B	42.3	42.5	48.3	44.7	33.2	49.0	20.0	45.6	45.3	39.6
	14B	35.9	35.1	39.7	38.9	26.8	32.5	6.7	37.1	34.8	40.7
	8B	34.4	35.5	38.9	37.2	26.7	36.7	5.7	37.3	41.3	24.3
GLM	4.1V-9B-Thinking	34.6	36.2	38.7	35.3	27.3	25.0	50.0	24.9	32.6	20.0

global counter ($\mathcal{G}_{\text{MRA}} = 71.7$) remains about 18 points below humans, and several open-weight models sit at or barely above the majority baseline. The only scores approaching the human range are the two G3.x models on synthetic AB (73.0, 71.8); we treat these as an anomaly, since the same two models are mid-pack on real attribute binding (45.5, 46.5) and no other model shows the effect. The self-preference check in Appendix A.2 establishes only that the generating model, G2.5-Pro, is not advantaged on its own images; neither G3.x model generated the synthetic split, so that control does not account for their gap. Across alternative option sets, shuffled answer options and deliberately biased option configurations, their synthetic performance moves by at most 3 points, so the effect is robust to the tested perturbations but remains an observed anomaly rather than evidence of genuine capability. We discuss more in Appendix A.2.

Attribute binding is far from being solved, and the weakest models barely beat chance predictors. On real images the ceiling is low: no model exceeds 52.8% $\mathcal{I}_{\text{ACC}}$ (GPT-5-medium), with G2.5-Flash (51.3) and GLM-4.5V (50.3) close behind—all far below the human baseline (Table 2). The only scores that rise much higher are the two G3.x models on synthetic (73.0, 71.8), which we flag as anomalous since those same models are mid-pack on real (discussed below). Against a random-choice baseline of 15.8% and a majority-answer baseline of 34.3% (on both the real and synthetic splits), the gains are modest: several models (Q3-30B-A3B-Instruct on both splits, GPT-5-minimal on both splits) fall at or near the majority baseline, adding little over always predicting the most frequent answer.

Numerical Counting requires forming the correct target set from an RE, excluding same-category distractors, and enumerating valid instances. Table 2 shows models differ in whether they favor global ($\mathcal{G}_{\text{MRA}}$) or

localized ($\mathcal{L}_{\text{MRA}}$) counting. G2.5-Pro attains the highest real-split $\mathcal{G}_{\text{MRA}}$ (71.7), while on the synthetic split the G3.x models lead (67.2 for G3.5-Flash-Low, 64.1 for G3.1-Pro-Preview-Low). Several locally evaluated open-weight models benefit from localized decomposition, e.g. IVL3-38B ($\mathcal{L}_{\text{MRA}}$=49.0 vs. $\mathcal{G}_{\text{MRA}}$=20.0 real) and IVL3.5-8B (45.8 vs. 13.3 real), suggesting that reducing the search space aids target-set formation when full-image counting is hard. A parent-level gap analysis (Appendix C) confirms this is not an aggregation artifact, and balanced success across both remains rare. Real images are harder mainly for instance-level and localized counting: G2.5-Pro drops from $\mathcal{I}_{\text{MRA}}$=56.1/$\mathcal{L}_{\text{MRA}}$=53.2 (synthetic) to 37.3/24.2 (real), with Q3-32B-Instruct showing a similar decline. NC trends are not explained by scale, as gains are non-monotonic across Qwen and InternVL variants.

Both AB and NC in dense same-category scenes remain far from solved. The gap to humans is largest on real images, where the strongest models trail substantially on both tasks.

MIMO-Crop. Most existing benchmarks (Tong et al., 2024; Chen et al., 2024a; Fu et al., 2024; Liu et al., 2024; Li et al., 2024) use low/standard-resolution images with sparsely distributed objects. MIMO-Bench uses comparatively high-resolution real (2500×1500) and synthetic (1024×1024) images, because high resolution allows a single scene to hold many same-category instances as distractors. We additionally evaluate models on MIMO-Crop, obtained by cropping a window around each target from the original images (mechanism in the appendix), in two variants: 512×512 and 768×768. Because each crop is a window rather than a downsample, the target keeps full fidelity and is enlarged relative to the frame, while fewer same-category distractors fall inside it. MIMO-Crop thus approximates the low/standard-resolution regime of existing benchmarks and lets us check whether the MIMO difficulty persists once resolution, object scale, and distractor burden are partially controlled.

Table 3: $\mathcal{I}_{\text{ACC}}$ on MIMO-Bench and MIMO-Crop 512/768, and standard low-res benchmarks. †Mean of MM-Star, RealWorldQA, and MM-Bench. I: Instruct variant.

Model	MIMO Bench	Crop 768	Crop 512	Existing Avg.†
Q3-32B-I	47.5	55.0	48.9	81.4
Q3-30B-A3B-I	33.8	42.2	44.9	77.3
Q3-8B-I	40.3	48.9	44.4	75.6
Q2.5-72B-I	42.6	45.7	48.3	78.3
Q2.5-32B-I	39.3	48.5	45.0	74.7
Q2.5-7B-I	35.8	42.8	47.6	71.7
IVL3.5-38B	42.8	51.2	55.5	79.5
IVL3.5-14B	34.4	46.9	46.4	74.1
IVL3.5-8B	40.5	48.4	50.7	72.1
IVL3-78B	39.3	47.9	49.5	79.4
IVL3-38B	44.3	52.8	56.7	78.0
IVL3-14B	37.2	51.1	50.7	74.3
IVL3-8B	35.9	44.7	46.8	73.6

Table 3 evaluates whether MIMO-Crop accounts for the gap between MIMO-Bench and existing low/standard-resolution benchmarks. Models improve on MIMO-Crop, confirming that full-image resolution, object scale, and distractor burden contribute to the difficulty. However, the recovery is incomplete: models improve from 39.5 on full MIMO-Bench to 48.2 on MIMO-Crop768 and 48.9 on MIMO-Crop512 on average, but remain far below their average performance on standard low/standard-resolution benchmarks (76.2). This shows that MIMO-Bench difficulty is not explained by resolution and object scale alone, as models struggle even when the target region is enlarged and the number of same-category distractors is reduced.

Synthetic gains are model-dependent The advantage is largest by far for the two G3.x models (+27.5 and +25.3 points on synthetic AB), with a more moderate gain for Q2.5-72B-Instruct (+12.0). Several other models, however, are flat or real-favored, including Q3-32B-Instruct (−0.7), GLM-4.5V (−8.6), G2.5, and GPT-5. This spread suggests synthetic scenes are not uniformly easier; rather, the synthetic-real gap reflects model-specific strengths rather than a general property.

Counting as a Function of Target-Set Size. We analyze numerical-counting performance as a function of the ground-truth (GT) target-set size. For each prediction $\hat{y}_i$ with GT count y_i, we compute signed error $e_i = \hat{y}_i - y_i$, bias $\frac{1}{n}\sum_i e_i$, MAE $\frac{1}{n}\sum_i |e_i|$, exact match ($\hat{y}_i = y_i$), off-by-one accuracy ($|\hat{y}_i - y_i| \leq 1$), and off-by-two-or-more error ($|\hat{y}_i - y_i| \geq 2$). +ve bias indicates overcounting, while -ve bias indicates undercounting. Full count-conditioned statistics are reported in the appendix.

The count-conditioned profile aggregated across models reveals a structure that is hidden by aggregate counting metrics. Sparse target sets are prone to false positives: for GT count 0, models achieve only 0.3333 exact match and exhibit a positive bias of +1.0833, indicating that they often count visually similar distractors as valid targets. Small-to-moderate target sets remain the most reliable regime overall: mean $\mathcal{MRA}$ reaches 0.4968 at count 3 and 0.5234 at count 5, while exact match in these bins is still only 0.3541 and 0.2888, respectively, indicating that many predictions are near misses rather than exact counts. This behavior

is analogous to the small-numerosity regime in cognitive science, where small sets can be rapidly individuated, while larger sets require slower serial enumeration or approximate numerosity estimation (Revkin et al., 2008; Piazza et al., 2011). As the target set becomes dense, performance degrades sharply and the error direction flips. At GT count 4, the average bias becomes negative (−0.5406), and undercounting grows with target-set size: the bias reaches −2.2774 at count 7, −3.1172 at count 10, −4.5490 at count 11, and −9.5556 at count 14. Exact match also collapses, dropping to 0.0292 at count 7 and 0.0000 at count 14. This mirrors cognitive evidence that spatially grouped, connected, or visually similar items can be underestimated because they are encoded as fewer perceptual units (Chakravarthi et al., 2023; Anobile et al., 2017). Taken together, these results suggest that counting errors are not uniform across target-set sizes: sparse scenes are vulnerable to false inclusion, whereas dense scenes are more often compressed into underestimates.

In InternVL3, language-model scaling alone does not explain AB gains. InternVL3 improves from the 8B/14B variants to 38B (real 34.4/35.9 → 42.3; synthetic 38.9/39.7 → 48.3), coinciding with an upgrade from the ∼300M-parameter InternViT encoder to InternViT-6B at ≥38B (alongside a larger language model). Beyond 38B the trend does not continue: IVL3-78B regresses to 40.2 (real) / 37.6 (synthetic) despite retaining the 6B encoder and using a far larger language model, suggesting that language-model scaling alone does not explain the observed gains.

Attribute-Level Failure Modes. Most object categories fail within a narrow band of 55–68% (person 63.3%, plant 64.1%, utensil 68.0%), with two-wheelers the exception at 36.8%. Coarse attributes ask what an object is or whether a part is present (type, presence), while fine-grained attributes ask for a finer property such as color, pattern, sleeve length, shape, material, or texture. Fine-grained attributes fail more often than coarse ones (65.5% vs. 55.7%). The gap is clearest when the same object is asked at both levels: a pillow's type fails 26.2% but its color 78.0% and pattern 75.3%; a bag's type fails 35.1% but its pattern 79.2%. Difficulty is uneven within the fine-grained group: pattern and color are consistently the hardest (pattern fails 70.9% across clothing, bags, and pillows), whereas shape, material, and sleeve length are more moderate and at times approach coarse-attribute rates (e.g., pillow shape 32.1%, plant-pot material 42.6%).

Note: Subsequent experiments in this and the next sections are done on an 800-sample MIMO-Bench Attribute Binding subset.

Thinking vs. Non-Thinking. Test-time reasoning does not uniformly improve MIMO performance (Table 4, as tested on a subset). Its clearest benefit appears in localized, decomposable counting. For Q3-8B, Thinking improves all reported metrics, with $\mathcal{I}_{\text{ACC}}$ increasing 39.2 → 44.4 and $\mathcal{L}_{\text{MRA}}$ increasing sharply 40.0 → 62.3. Q3-30B-A3B also improves on both attribute-binding metrics ($\mathcal{I}_{\text{ACC}}$ 35.2 → 37.0, $\mathcal{G}_{\text{ACC}}$ 32.3 → 37.3). In contrast, Q3-32B shows a mixed effect: Thinking slightly reduces AB ($\mathcal{I}_{\text{ACC}}$ 46.3 → 46.2) and global counting ($\mathcal{G}_{\text{MRA}}$ 50.4 → 44.2), while still improving localized counting ($\mathcal{L}_{\text{MRA}}$ 46.1 → 48.9). A similar lack of uniformity appears in proprietary models: higher reasoning effort largely hurts G3.1-Pro-Preview on counting ($\mathcal{I}_{\text{MRA}}$ 62.5 → 57.7, $\mathcal{L}_{\text{MRA}}$ 58.7 → 52.8) but helps G3.5-Flash across numerical-counting metrics ($\mathcal{I}_{\text{MRA}}$ 60.3 → 66.7, $\mathcal{G}_{\text{MRA}}$ 58.3 → 63.3, $\mathcal{L}_{\text{MRA}}$ 58.3 → 71.6). Overall, test-time reasoning does not reliably improve performance on MIMO-Bench.

Table 4: **Effect of thinking/reasoning.** ("w/o","w/") denotes (Instruct/Thinking) variant for Qwen and (Low/High) reasoning effort for Gemini.

Model	Thk.	$\mathcal{I}_{\text{ACC}}$	$\mathcal{G}_{\text{ACC}}$	$\mathcal{I}_{\text{MRA}}$	$\mathcal{L}_{\text{MRA}}$	$\mathcal{G}_{\text{MRA}}$
Q3-8B	w/o	39.2	40.1	43.9	40.0	37.3
	w/	44.4	44.8	47.3	62.3	43.2
Q3-30B-A3B	w/o	35.2	32.3	34.0	39.7	30.5
	w/	37.0	37.3	35.4	40.1	31.3
Q3-32B	w/o	46.3	46.0	46.3	46.1	50.4
	w/	46.2	43.5	48.8	48.9	44.2
G3.1-Pro-Preview	w/o	53.8	54.2	62.5	58.7	62.3
	w/	53.2	54.3	57.7	52.8	63.5
G3.5-Flash	w/o	53.3	56.5	60.3	58.3	58.3
	w/	56.9	54.6	66.7	71.6	63.3

5 Complexity Breakdown

Model's error on attribute binding arises from two major distinct stages: the difficulty of comprehending the referring expression (RE) in order to reach the target, and the difficulty of binding attributes once the target is found. Both are affected by the presence of similar-feature distractors. Hence, we ask:

- How do MLLMs perform when the complexity of comprehending the referring expression is removed?
- How do MLLMs perform when the presence of distractors is reduced?

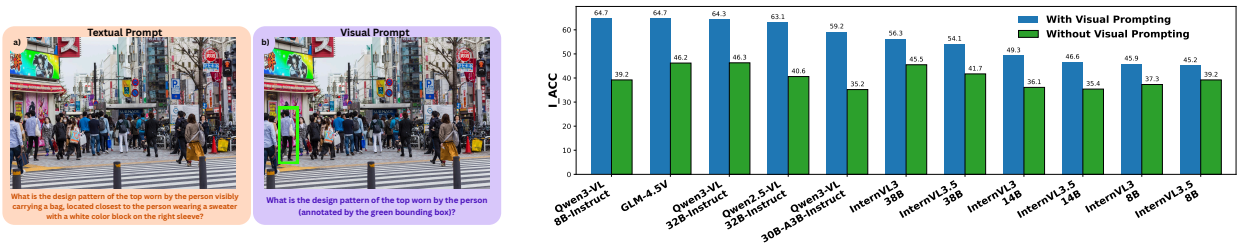


Figure 5: Visual prompting setup. Highlighting the target with a bounding box (a) instead of an RE consistently improves $\mathcal{I}_{\text{ACC}}$ (b).

Visual Prompting. We test how models perform when the referring expression no longer has to be comprehended. In the standard setting the model must parse the textual RE and ground it to the correct instance; in the visual-prompting setting we remove this step by replacing the RE with a bounding box/polygon drawn directly on the target. Fig. 5 illustrates how an image–question pair transforms under visual prompting, and reports the effect: removing the RE improves $\mathcal{I}_{\text{ACC}}$ for every evaluated model. The benefit also grows with RE complexity. As shown in Fig. 6, the gain from visual prompting increases as the number of compositional elements in the expression rises, indicating that the RE is a larger obstacle precisely when it is most compositional. Even with the target marked directly, **no model exceeds ∼65%**, though the marker alters the image, so the residual error cannot be attributed to attribute binding alone; we attempt to disentangle grounding from attribute binding in Appendix C.5.1.

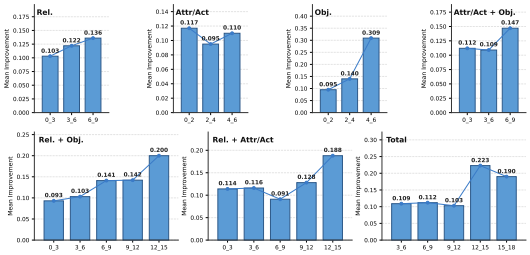


Figure 6: Effect of visual prompting on $\mathcal{I}_{\text{ACC}}$.

Replacing the referring expression with a visual prompt raises $\mathcal{I}_{\text{ACC}}$ by +6.0 to +25.5 points (mean +15.5) for every evaluated model, yet none exceeds ∼65%.

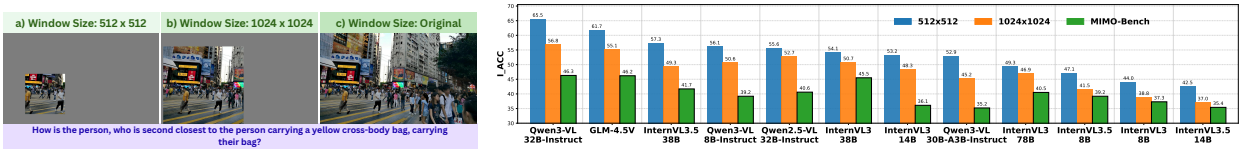


Figure 7: Visual Distractors Effect: (left) Graying setup & (right) Performance trends as distractors increase.

Performance as a Function of Visual Distractor Reduction. We study how model performance changes under interventions that reduce visual distractor burden. We first analyze this through graying: we gray out the entire image except for a visible window centered on the queried object, using window sizes of 512 and 1024, where smaller windows leave fewer same-category distractors visually active (Fig. 7). Throughout all configurations, the target object's scale, image resolution, and textual prompt remain fixed. Fig. 7 shows that $\mathcal{I}_{\text{ACC}}$ generally rises as the window shrinks and fewer distractors remain in view, across the majority of evaluated models. The effect is directional but not strictly proportional: as detailed in the appendix, the accuracy gained per distractor removed varies substantially across images and even between window sizes of the same image, so **distractor count alone does not predict the size of the change**.

Graying is a valid distractor-reduction intervention, but not a clean causal isolation of distractor count alone: it reduces visual competition from surrounding objects by suppressing the surrounding region, while also suppressing surrounding context and increasing the relative salience of the retained region. We therefore compare it against blur and inpainting controls in Appendix C.6. Across these interventions, performance generally improves as distractor burden is reduced, showing that the directional effect is not specific to graying. Inpainting provides the closest practical approximation to directly removing selected same-category distractors while preserving the rest of the image, although segmentation and generation artifacts prevent treating it as an exact counterfactual. **The similar behavior observed under inpainting strengthens the interpretation that reducing visual distractor burden benefits MIMO performance, while we do not claim a clean causal effect of distractor count alone.**

The two manipulations help different models. From Figs. 5 and 7, models such as Q3-8B-Instruct, Q2.5-32B-Instruct, Q3-30B-A3B-Instruct, and GLM-4.5V gain more from removing the referring expression, whereas others (e.g., IVL3.5-38B, IVL3-14B, and Q3-32B-Instruct) gain more from reducing distractors.

6 What Enhances MLLM Performance in MIMO?

What helps in performing well in MIMO? A human analogy. In MIMO settings, objects rarely exhibit a clear visual "pop-out" effect, requiring humans to actively search the scene to locate targets. This search is typically guided by semantic and syntactic cues from referring expressions or internal reasoning, combined with pre-attentive visual features (Wolfe et al., 1989; Treisman & Gelade, 1980). When such guidance is weak, humans often resort to a serial search strategy, scanning objects sequentially across the scene. Effective human perception therefore involves examining the image *dynamically* at multiple spatial scales while iteratively refining attention rather than fixed at input, as in a static forward pass.

We hypothesize that this dynamic visual access is what the MIMO setting rewards. Notably, several top-performing MLLMs on MIMO-Bench already incorporate such mechanism: Qwen3-VL's DeepStack (Bai et al., 2025a) re-injects visual features across multiple layers of the language model, enabling access to visual representations at different levels of abstraction, while GPT-style interleaved image–text reasoning (OpenAI, 2025a) repeatedly re-attends to relevant image regions across intermediate reasoning steps. We cannot directly verify that these mechanisms are responsible for the observed performance. To measure the effect of dynamism, we perform a small exploratory study.

We perform a multi-turn inference in which the model is posed with the original question and must propose a region – specified as bounding-box coordinates – that it deems sufficient to answer the question. This region is cropped from the image, and the cropped image is re-queried together with the original question to obtain the final answer. We refer to this as *naive* relevant-region extraction as the model is not explicitly trained to propose a relevant crop. As shown in Table 5, even this naive single-pass narrowing yields a consistent improvement across all evaluated models, supporting the hypothesis that dynamic narrowing of the search space benefits performance on MIMO-Bench, akin to the human mechanism. We further include fixed/random crop controls, a two-pass no-crop control, and IoU-conditioned accuracy to disentangle localization quality from cropping, rescaling, and the additional inference pass; see Appendix C.7.

Table 5: Effect of zooming on model performance. Δ denotes the change in $\mathcal{I}_{\text{ACC}}$ relative to MIMO-Bench.

Model	Zoom.	Δ
Q3-8B-Instruct	Naive	+0.89
Q2.5-7B-Instruct	Naive	+1.50
IVL3-8B	Naive	+5.46
IVL3-14B	Naive	+16.90
IVL3.5-8B	Naive	+2.14
IVL3.5-14B	Naive	+11.75
Q3-32B-Instruct	Naive	+3.00

Letting a model narrow the search space before answering consistently improves accuracy, indicating that MIMO rewards dynamic visual access rather than a single static forward pass.

Does CoT Enhance MIMO Performance? We evaluate whether text-based Chain-of-Thought (CoT) prompting helps performance on MIMO-Bench. Following prior prompting setups (Yang et al., 2025), we test Zero-Shot CoT (ZS-CoT) and Self-Consistency CoT (SC-CoT). For SC-CoT, we run five CoT generations and use majority voting over the predicted answers. As shown in Fig. 8, CoT yields model-dependent gains rather than consistently improving performance. Several models benefit from at least one CoT variant: IVL3.5-8B improves from 0.39 to 0.46 under ZS-CoT, IVL3-14B from 0.36 to 0.42 under SC-CoT, and Q3-8B from 0.39 to 0.43 under both variants. Q2.5-7B improves marginally from 0.36 to 0.37 under SC-CoT, while IVL3.5-14B remains nearly unchanged. In contrast, several models degrade under CoT: IVL3-8B slightly decreases, Q2.5-32B drops from 0.41 to 0.39 with ZS-CoT and 0.38 with SC-CoT, and Q3-32B drops from 0.46 to 0.44 with ZS-CoT and 0.40 with SC-CoT. These results suggest that CoT can provide modest gains for some models, but is not uniformly beneficial and may degrade performance for others.

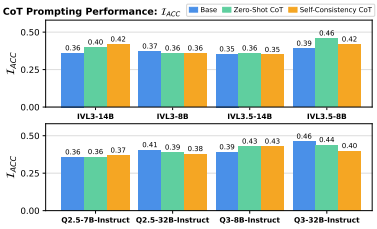


Figure 8: CoT on MIMO-Bench.

7 Conclusion

We introduce MIMO-Bench, a benchmark for evaluating multimodal LLMs in settings with multiple instances across multiple object categories. MIMO-Bench is constructed around the complexity introduced by distractors, and our experiments show how strongly distractors hinder performance. We further show that removing the visual subtask posed by referring expressions yields a large performance gain, while our distractor-reduction interventions show that reducing visual distractor burden generally improves model performance. Based on these findings, we believe current MLLMs should first master the elementary tasks of attribute binding and numerical counting in MIMO before tackling harder problems. We also argue that, in settings such as MIMO, an ideal evaluation should couple reasoning with localization, requiring models to localize the referred object. We explored this through two interfaces: selecting the correct target among visually marked candidate boxes, and directly predicting its coordinates. These interfaces impose different demands and yield substantially different localization behavior (Appendix C.5.1), preventing us from treating localization as a stable mandatory intermediate benchmark stage. Such localization can be viewed as an additional feature alongside semantic features, making the binding task more complete. We release the target masks and boxes to facilitate future evaluation along this direction. As models become more robust, we hope MIMO-Bench serves as a realistic benchmark and further inspires research on improving reasoning in multimodal LLMs.

References

Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. In *AAAI*, pp. 8076–8084, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, pp. 23716–23736, 2022.

Giovanni Anobile, Guido Marco Cicchini, Antonella Pomè, and David C. Burr. Connecting visual objects reduces perceived numerosity and density for sparse but not dense patterns. *Journal of Numerical Cognition*, 3(2):133–146, 2017. doi: 10.5964/jnc.v3i2.38.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL `https://arxiv.org/abs/2308.12966`.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. `https://arxiv.org/abs/2511.21631`, 2025a.

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

Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M. Frankland, Thomas L. Griffiths, Jonathan D. Cohen, and Taylor Webb. Understanding the limits of vision language models through the lens of the binding problem. In *NeurIPS*, pp. 113436–113460, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *ICLR*, 2026. URL `https://openreview.net/forum?id=r35clVtGzw`.

Ramakrishna Chakravarthi, Andy Nordqvist, Marlène Poncet, and Nika Adamian. Fundamental units of numerosity estimation. *Cognition*, 239:105565, 2023. doi: 10.1016/j.cognition.2023.105565.

Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S.-H. Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. In *CVPRW*, pp. 513–524, 2025.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, pp. 27056–27087, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pp. 24185–24198, 2024b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Robert Dale and Ehud Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.

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 *ICLR*, 2026.

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 *ECCV*, pp. 148–166, 2024.

GLM-V Team, Wenyi Hong, Wenmeng Yu, et al. GLM-4.1V-Thinking and GLM-4.5V: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Google DeepMind. Gemini 3.1 Pro preview. `https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview`, 2026a. Model release / documentation page. Accessed: 2026-07-03.

Google DeepMind. Gemini 3.5 Flash. `https://ai.google.dev/gemini-api/docs/models/gemini-3.5-flash`, 2026b. Model release / documentation page. Accessed: 2026-07-03.

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

H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and Semantics, Vol. 3: Speech Acts*, pp. 41–58. Academic Press, New York, 1975.

Xiangzhao Hao, Kuan Zhu, Hongyu Guo, Haiyun Guo, Ning Jiang, Quan Lu, Ming Tang, and Jinqiao Wang. Referring expression instance retrieval and a strong end-to-end baseline. In *ACM MM*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *ACL*, pp. 3828–3850, 2024.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In *NeurIPS*, pp. 31096–31116, 2023.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pp. 6693–6702, 2019.

Amirmohammad Izadi, Mohammad Ali 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 *NeurIPS*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ee8351f13c3015ac249ed5f767096966-Abstract-Conference.html.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pp. 2901–2910, 2017.

Edna L Kaufman, Miles W Lord, Thomas Whelan Reese, and John Volkmann. The discrimination of visual number. *The American Journal of Psychology*, 62(4):498–525, 1949.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, pp. 787–798, 2014.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pp. 4015–4026, 2023.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. NaturalBench: Evaluating vision-language models on natural adversarial samples. In *NeurIPS*, pp. 17044–17068, 2024.

Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, pp. 34892–34916, 2023b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *ECCV*, pp. 216–233, 2024.

Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, Benoit Dumoulin, and Hanghang Tong. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in VLMs. In *ICLR*, 2026.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/663bce02a0050c4a11f1eb8a7f1429d3-Abstract-Conference.html.

Guangtao Lyu, Xinyi Cheng, Chenghao Xu, Qi Liu, Muli Yang, Fen Fang, Huilin Chen, Jiexi Yan, Xu Yang, and Cheng Deng. Revealing perception and generation dynamics in LVLMs: Mitigating hallucinations via validated dominance correction. *arXiv preprint arXiv:2512.18813*, 2025.

Guangtao Lyu, Xinyi Cheng, Qi Liu, Chenghao Xu, Jiexi Yan, Muli Yang, Fen Fang, and Cheng Deng. Towards interpretable hallucination analysis and mitigation in LVLMs via contrastive neuron steering. *arXiv preprint arXiv:2602.00621*, 2026a.

Guangtao Lyu, Qi Liu, Chenghao Xu, Jiexi Yan, Muli Yang, Xueting Li, Fen Fang, and Cheng Deng. Revealing and enhancing core visual regions: Harnessing internal attention dynamics for hallucination mitigation in LVLMs. In *Findings of ACL*, pp. 15214–15230, 2026b.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. CREPE: Can vision-language foundation models reason compositionally? In *CVPR*, pp. 10910–10921, 2023.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pp. 11–20, 2016.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *WACV*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *WACV*, pp. 1697–1706, 2022.

OpenAI. Thinking with images. `https://openai.com/index/thinking-with-images/`, 2025a.

OpenAI. GPT-5 system card. `https://openai.com/index/gpt-5-system-card/`, 2025b.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to count to ten. In *ICCV*, pp. 3170–3180, 2023.

Manuela Piazza, Antonia Fumarola, Alessandro Chinello, and David Melcher. Subitizing reflects visuo-spatial object individuation capacity. *Cognition*, 121(1):147–153, 2011.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, pp. 3394–3403, 2021.

Susannah K. Revkin, Manuela Piazza, Véronique Izard, Laurent Cohen, and Stanislas Dehaene. Does subitizing reflect numerical estimation? *Psychological Science*, 19(6):607–614, 2008.

Adina L Roskies. The binding problem. *Neuron*, 24(1):7–9, 1999.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, 2024. doi: 10.52202/079017-2771. URL `https://proceedings.neurips.cc/paper_files/paper/2024/hash/9ee3a664ccfeabc0da16ac6f1f1cfe59-Abstract-Conference.html`.

Anne Treisman and Hilary Schmidt. Illusory conjunctions in the perception of objects. *Cognitive Psychology*, 14(1):107–141, 1982.

Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12 (1):97–136, 1980.

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

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

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*, pp. 7907–7915, 2025b.

Jeremy M. Wolfe, Kyle R. Cave, and Sarah L. Franzel. Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 15(3):419–433, 1989.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *CVPR*, pp. 13084–13094, 2024.

xAI. RealWorldQA: A multimodal benchmark for real-world spatial understanding. https://x.ai/news/grok-1.5v, 2024. Released alongside the Grok-1.5V model.

Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions. In *NeurIPS*, 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 *CVPR*, pp. 10632–10643, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *NeurIPS*, pp. 21875–21911, 2024.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pp. 69–85, 2016.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/df29d63af05cb91d705cf06ba5945b9d-Abstract-Conference.html.

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.

Appendix

In this appendix, we discuss

- Benchmark construction including image curation, the synthetic-image self-preference check, the SAM3 distractor-count protocol, QA generation, attribute dictionary construction, referring expression parsing, the human annotation protocol, and scalability (Appendix A).
- Creation of MIMO-Crop (Appendix B).
- Additional experimental details and results, including evaluation setup, strict Exact Group Accuracy, the parent-level global–local counting comparison and hierarchical consistency, full count-conditioned counting statistics and per-count support, the visual prompting and grounding analyses, the graying and relevant-region controls, and uncertainty and significance analysis (Appendix C).
- Inference mechanisms for naive relevant-region extraction (Appendix C.7) and for zero-shot and self-consistency CoT (Appendix C.8).
- Limitation of the current work (Appendix D).

A Benchmark Construction

In this section, we describe the complete dataset and the benchmark generation process. Our benchmark comprises two tasks: *attribute binding* and *numerical counting*. Attribute binding tasks are defined with respect to a specific target instance. We consider VQA questions that consist of a primary task pertaining to a target object. Consequently, each target object is referenced by a referring expression that must uniquely localize the object in the image. For example, an attribute binding question may be of the form: "What is the color of the *car*?" where the italicized phrase serves as the referring expression.

For numerical counting, the question similarly relies on a referring expression to specify the set of instances to be counted. For example, "Count the number of *glass jars* in the image?" where the referring expression identifies the exact group of objects to be counted.

A.1 Image Curation

For synthetic images, we use G2.5-Pro (Comanici et al., 2025) for image generation. Example prompts and their corresponding generated images are shown in Fig. 9. Although the prompts requested a resolution of 2000×1500 pixels, G2.5-Pro returned images at its default resolution of 1024×1024 pixels. We reproduce the prompts verbatim, including the requested resolution, and retain all synthetic images at the returned 1024×1024 resolution. As each image is generated, we inspect the physical content of the image. We ensure that the image does not have inconsistencies such as color smudges (color from an object spreading into the background), physically disconnected objects (e.g., leaves from a plant are disconnected and seem to be hanging in the air), objects with impossible interactions (the leg of a table passing through the chair seat), etc.

We sample natural images from the SA-1B (Kirillov et al., 2023) dataset using two filters. First, since MIMO-Bench requires object-dense scenes with many candidate distractors, we filter images using the number of available SA-1B mask/box annotations and discard images with insufficient annotated objects. Second, among the remaining object-dense images, we apply a depth-based filter to avoid close-up or visually flat scenes. Specifically, we estimate metric depth using the outdoor variant of Depth Anything V2 (Yang et al., 2024) and prioritize images with larger estimated scene depth. Images are retained only if they pass both the object-density filter and the depth filter. This depth score is used only as a sampling heuristic; final inclusion is determined by manual inspection, retaining images that contain multiple instances of at least one object category and support visually grounded, unambiguous referring expressions.

To quantify the same-category distractor density of the curated images, we compute per-category instance counts using SAM3 (Carion et al., 2026), a promptable concept segmentation model. For each image and each



Figure 9: Example prompts and corresponding generated synthetic images for MIMO-Bench. We consistently swap aesthetic details to gather varying images characterized by the MIMO setting.

annotated object category, we supply the category name (e.g., “person”, “plant”, “vase”) as a text concept prompt; SAM3 returns one instance-level segmentation mask per detected instance, and the distractor count is the total number of returned masks. These counts underlie the per-category instance-count distributions reported in Fig. 2 of the main paper.

Data availability. We will release all MIMO-Bench annotations and evaluation code. For the real-image split, we will release the corresponding SA-1B image identifiers and reconstruction instructions; users must



Figure 10: **Qualitative comparison of synthetic images generated by Grok-4, Flux, and G2.5-Pro** under the same MIMO-style prompts. G2.5-Pro produces more realistic, feature-diverse scenes with stronger prompt adherence. Grok-4 and Flux exhibit reduced realism, weaker feature diversity, and object-level artifacts, making them less suitable for constructing unambiguous MIMO-style QA pairs.

obtain the underlying images directly from Meta under the SA-1B Dataset Research License, and we will not redistribute those images. For the synthetic split, we will release the Gemini-2.5-Pro-generated images together with their annotations under the terms specified in the accompanying repository.

A.2 Synthetic Image Self-Preference Check

Since the synthetic images in MIMO-Bench are generated using G2.5-Pro (Comanici et al., 2025), we additionally check whether G2.5-Pro obtains an artificial advantage on this split. We compare G2.5-Pro with proprietary and open-source models on a small synthetic attribute-binding pilot using the same prompt, answer-format constraints, and decoding setup as the main benchmark evaluation. As shown in Table 6, G2.5-Pro does not show a clear self-preference advantage on Gemini-generated images: its performance is comparable to strong non-Gemini models rather than being uniquely dominant. This suggests that the synthetic split does not merely favor the image generator, and that performance on this split still reflects the ability to ground compositional referring expressions and bind attributes under dense same-category distractors. This control covers only G2.5-Pro: neither G3.5-Flash nor G3.1-Pro-Preview generated the synthetic split, so applying the same check to them would test family-level affinity rather than generator self-preference. We therefore probed the G3.x synthetic result directly for evaluation artifacts, re-running the synthetic attribute-binding split under alternative option sets, shuffled answer options and deliberately biased option configurations; synthetic performance remained within 3 points in every condition. These checks rule out the tested option-ordering and answer-set artifacts, but do not rule out evaluation artifacts in general and do not explain the gap, which we therefore report as an observed anomaly rather than genuine capability or a generator effect.

Model	**Syn. AB** $\mathcal{I}_{\text{ACC}}$
G2.5-Pro	47.7
GPT-5-medium	49.7
IVL3.5-241B-A28B	45.1
Q3-32B-Instruct	47.9

Table 6: **Self-preference check on Gemini-generated images.** Attribute-binding instance accuracy on the synthetic split. G2.5-Pro does not exhibit a clear self-preference advantage despite being used to generate the synthetic images.

We also qualitatively compared G2.5-Pro with Grok-4 and Flux for synthetic image generation. Although all three models could generate dense multi-instance scenes, G2.5-Pro was selected because it more reliably produced visually coherent scenes with stronger prompt adherence. In comparison, Grok-4 and Flux more frequently produced images with reduced realism, weaker feature diversity, or object-level artifacts that made them less suitable for constructing unambiguous MIMO-style QA pairs. Fig. 10 provides a qualitative side-by-side illustration.

A.3 QA Generation

Each object in an image is annotated by referring expression(s) and bounding polygons. Alongside this, images for numerical counting are annotated on a group-wise basis, as detailed in Sec. 4.2 and illustrated in Fig. 4 of the main paper.

For attribute binding, each object in an image is mapped with its attribute dictionary to generate an exhaustive set of raw *multiple-choice* image-question pairs with single-choice answers. For numerical counting, the annotations are prefixed by the phrase "Count the number of . . . ", and the expected answer is a number. These raw image-question pairs are then answered and verified by humans.

Considerations:

Few attributes, such as color, are continuous. We construct multiple-choice options using carefully selected negative categories. For example, when the ground-truth color is "blue", we exclude candidate colors that are perceptually too similar to blue to avoid overlap or ambiguity (see id 4 in Fig. 20). Instead, we select visually distinct and commonly used color terms as answer options.

A.4 Attribute Dictionaries

Attribute Dictionaries are hierarchical dictionaries that contain the possible attributes of an object category or its components could possess. The discrete attributes usually have a set of plausible answers which are exclusive and exhaustive, whereas continuous attribute values have a soft boundary and are left to the human answering annotator to decide what the value should be. These dictionaries are mapped with target objects from an image and passed through a template-based question generation script that generates a raw list of multiple-choice questions. We join the prompts from Figs. 11, 12(represented in two figures due to spacing limitations) in order to generate the dictionaries for different object categories Figs. 13, 14, 15, and 16 present attribute dictionaries corresponding to "person", "plant", "lamp", and "utensil", respectively.

A.5 Referring Expression Parsing

As discussed in the main paper, each referring expression is made up of attributes/actions, relations, and neighboring objects. To parse it, we use G2.5-Pro (Comanici et al., 2025). We use the prompts as shown in Figs. 17, 18a, and 18b and append them with the expression we want to parse. Each output from the model is stored as shown in the structure in the prompting example. Parsed expressions help us to disentangle the causes of certain results.

The parser converts a referring expression into a structured scene graph consisting of: (i) **objects**, base noun phrases referenced in the expression (lemmatized, lowercase); (ii) **Attr** (the `cao` field in the parser output), attribute/action/state bindings that modify a single object (e.g., color, material, or an intransitive action such as "walking"); (iii) **coro**, object–relation–object triples linking two objects (transitive actions, spatial/topological relations, possession); and (iv) **plain_objects**, objects that do not appear as the target of any Attr entry. Each parse also reports a `counts` field with the number of relations, Attr bindings, and plain objects.

We define the number of visual subtasks posed by a referring expression as the sum

$$\#\text{subtasks} = \#\text{relations} + \#\text{Attr} + \#\text{plain_objects}, \tag{4}$$

i.e., the count of `coro` triples plus the count of Attr bindings plus the count of plain objects returned by the parser for that expression. The mean subtask count reported for a benchmark (e.g., 9.6 for MIMO-Bench in Table 1) is the average of Eq. 4 over all parsed referring expressions in that benchmark. The parser and this counting rule are benchmark-agnostic: the same prompt and formula, applied to plain natural-language referring expressions with no benchmark-specific structure required, were used to obtain the subtask counts for MIMO-Bench and for every other benchmark listed in Table 1 (CLEVR, GQA, MMStar, NaturalBench, VQAv2, V*Bench, HR-Bench, and MME-RealWorld).

Table 7 reports these components across benchmarks. Object count alone does not distinguish MIMO-Bench, and CLEVR contains more Attr bindings; the clearest separation is in relations, where MIMO-Bench

Use the following definitions of attributes and components when reasoning about objects.

1. Object
  - An object is the main entity we are talking about (e.g., vase, person, plant).

2. Direct Attributes of an Object
  - Definition: Direct attributes are properties that can be asked directly about the object itself.
  - The question pattern is:
    "What is the <attribute> of the <object>?"
    Examples:
      - "What is the color of the vase?"
      - "What is the shape of this object?"
  - The answer must be:
    - Categorical (e.g., red, blue, square, round),
    - Verifiable from the image,
    - Objective (not opinion-based).

3. Components of an Object
  - Definition: Components are parts of an object.
  - For each object, we define component categories. For example, for a person:
    - t-shirt, pants, belt, cap, eyewear, footwear, etc.
  - Components are not separate independent objects; they are parts belonging to the main object.

4. Attributes of Components
  - Definition: These are properties of a component of an object, not of the whole object directly.
  - Each component can have relevant attributes, and each attribute has a value.
  - The question pattern is:
    "What is the <attribute> of the <component> of this <object>?"
    Examples:
      - "What is the color of the leaves of this plant?"
        -> component = leaves, object = plant, attribute = color.
      - "What is the color of the hair of the person?"
      - "What is the color of the t-shirt of this person?"
  - The value of an attribute must be valid for that attribute-component pair.
    Examples:
      - Valid: color(t-shirt) = "white"
      - Invalid: color(t-shirt) = "running" (not a color)

5. Discrete vs Continuous Attributes
  - In the schema below, an attribute whose value list is:
    - Empty list [] -> treated as a continuous / open-ended attribute (e.g., any color string, a numeric value, or a free-form but objective description).
    - Non-empty list [...] -> treated as a discrete categorical attribute with values restricted to the given options.

6. HighProb vs LowProb Groups
  - For some objects, attributes are organized into two groups:
    - "HighProb": attributes/components that are highly likely to be present in the scene (e.g., clothing and footwear for a visible person).
    - "LowProb": attributes/components that are less likely to be present in the scene (e.g., headwear, watch, certain bags).
  - These groups do NOT change the semantics of the attributes themselves; they only express how likely those components/attributes are to appear in a typical scene.

7. Example Hierarchical Attribute Schema (Person)
  - The following is an example of the hierarchical structure we expect.
  - It is only illustrative; the real schema may include more components and attributes.

Figure 11: Pre-prompt used for generating attribute-binding dictionaries for different object categories (1/2).

averages 3.23 against at most 1.34 elsewhere, roughly 2.4× the next highest. The elevated subtask count therefore comes mainly from relational structure rather than expression length.

```
{
 "Person": {
  "HighProb": {
   "Components": {
    "BaseLayerTop (first visible layer worn on the top)": {
     "Type": ["t-shirt", "shirt", "kurta"],
     "Color": [],
     "SleeveLength": ["full", "half/short", "sleeveless"],
     "Pattern": ["solid", "horizontally striped", "checkered"]
    }
    "Footwear": {
     "Type": ["sandals", "slippers", "laced shoes"],
     "Color": []
    }
   }
  },
  "LowProb": {
   "Components": {
    "Headwear": {
     "Type": ["cap", "hat", "helmet"],
     "Color": []
    },
    "Bag": {
     "Type": ["backpack", "handbag", "sling"],
     "Color": [],
     "CarryPosition": ["back", "left shoulder", "right shoulder", "cross-body", "left palm", "right palm"]
    }
    }
   }
  }
 }
}

Hierarchy recap:
 - Object level: "Person"
 - Probability tier: "HighProb" and "LowProb" (likelihood of being present in the scene)
 - Components: e.g., "BaseLayerTop", "LowerBodyClothing", "Footwear", "Bag", "Watch"
 - Each component has attributes (Type, Color, SleeveLength, etc.) where:
   - Attributes with [] are continuous / open-ended.
   - Attributes with ["val1", "val2", ...] are discrete categorical attributes.

INSTRUCTION TO THE MODEL:
- When reasoning about attributes, always distinguish between:
  (a) Direct attributes of the object, and
  (b) Attributes of the object's components.
- Use the appropriate question pattern:
  - Direct attribute: "What is the <attribute> of the <object>?"
  - Component attribute: "What is the <attribute> of the <component> of this <object>?"
- Interpret the schema as:
  - Non-empty lists define the allowed discrete categorical values.
  - Empty lists [] indicate continuous or open-ended attributes whose values are not restricted to a predefined set but must still be objective and
visually verifiable.
- Treat "HighProb" components/attributes as more likely to appear in the scene and "LowProb" components/attributes as less likely, but apply the
same reasoning rules to both.
- When in doubt, treat properties of parts (like leaves, hair, t-shirt, shoes, bags, watches) as attributes of components, not as direct attributes of the
whole object.
```

Figure 12: Pre-prompt used for generating attribute-binding dictionaries for different object categories (2/2).

Human


```
{
  "Person": {
    "HighProb": {
      "Components": {
        "BaseLayerTop (first visible layer worn on the top)": {
          "Type": ["t-shirt", "shirt", "polo", "kurta", "sweatshirt", "vest", "dress/frock", "top"],
          "Color": [],
          "SleeveLength": ["full", "half/short", "sleeveless", "3/4"],
          "Pattern": ["solid", "horizontally striped", "checkered", "printed", "vertically striped"]
        },
        "UpperBodyClothing (visible layers over the first/base visible layer of top)": {
          "Type": ["jacket", "sweater", "hoodie", "blazer", "cardigan", "overshirt"],
          "Color": [],
          "SleeveLength": ["full", "half/short", "sleeveless", "3/4"],
          "Pattern": ["solid", "horizontally striped", "checkered", "printed", "vertically striped"]
        },
        "LowerBodyClothing": {
          "Type": ["full length jeans", "trousers", "cotton shorts/capris", "skirt", "leggings", "denim shorts/capris
     ", "chinos", "linen pants", "sweatpants"],
          "Color": []
        },
        "Footwear": {
          "Type": ["sandals", "slippers", "barefoot", "boots(ankle or calf)", "knee boots", "laced shoes(includes
     sports shoes, sneakers)", "loafers(includes moccasins, slip-ons, loafers)", "heels"],
          "Color": []
        }
      }
    },
    "LowProb": {
      "Components": {
        "Headwear": {
          "Type": ["cap", "hat", "beanie", "turban", "helmet"],
          "Color": []
        },
        "Bag": {
          "Type": ["backpack", "handbag", "tote", "sling", "briefcase", "purse", "duffel bag"],
          "Color": [],
          "CarryPosition": ["back", "front", "left shoulder", "right shoulder", "left forearm", "right forearm", "
     cross-body", "left palm", "right palm", "around the waist"]
        },
        "Watch": {
          "Wrist": ["left", "right", "unknown"],
          "Color": []
        }
      }
    }
  }
}
```


Figure 13: Person attribute dictionary in JSON format.

Table 7: Referring-expression composition across benchmarks, parsed with the same benchmark-agnostic parser. Objects, Attr bindings and relations are the three terms of Eq. 4; same-category distractor density is reported in Table 1.

Benchmark	**Objects**	**Attr**	**Relations**	**# Subtasks**
CLEVR	2.50	**3.50**	1.20	7.20
GQA	2.24	0.42	0.43	3.10
MMStar	3.36	0.08	0.46	3.90
NaturalBench	1.87	0.53	0.30	2.70
VQAv2	1.76	0.35	0.49	2.60
V*Bench	1.92	0.07	0.40	2.40
HR-Bench	2.92	0.44	1.34	4.70
MME-RealWorld	2.70	0.25	0.95	3.90
MIMO-Bench	**4.01**	2.33	**3.23**	**9.56**

Plant


```
{
  "Plant": {
    "HighProb": {
      "DirectAttributes": {
        "Type": ["tree", "shrub", "herb", "creeper", "climber"]
      },
      "Components": {
        "Leaves": {
          "Color": [],
          "Shape": ["oval", "lanceolate", "heart-shaped", "needle-like", "lobed", "round"]
        },
        "StemOrTrunk": {
          "Color": ["green", "brown", "grey"],
          "Branching": ["unbranched", "branched"]
        }
      }
    },
    "LowProb": {
      "DirectAttributes": {
        "FlowerPresence": ["yes", "no"],
        "FruitPresence": ["yes", "no"]
      },
      "Components": {
        "Flowers": {
          "Color": []
        },
        "FruitsOrPods": {
          "Color": [],
          "Presence": ["present", "absent"]
        },
        "Roots": {
          "Visibility": ["exposed", "underground"],
          "Color": []
        },
        "PotOrContainer": {
          "Material": ["clay", "ceramic", "plastic", "metal", "cement"],
          "PrimaryColor": [],
          "Shape": ["round", "cylindrical", "rectangular", "conical/bowl"],
          "Texture": ["rough", "matte", "glossy", "ridged/groovy"],
          "Pattern": ["plain", "horizontally striped", "vertically striped", "dotted", "floral", "geometric", "
     checked/chequered", "gradient"],
          "StripedColors": {
            "Stripe1Color": [],
            "Stripe2Color": []
          }
        }
      }
    }
  }
}
```


Figure 14: Plant attribute dictionary in JSON format.

Lamp


```
{
  "Lamp": {
    "HighProb": {
      "DirectAttributes": {
        "Type": ["table lamp", "floor lamp", "wall-mounted lamp", "desk lamp", "hanging lamp"]
      },
      "Components": {
        "Lampshade": {
          "Shape": ["conical", "cylindrical", "drum", "bell", "square"],
          "Color": [],
          "Transparency": ["transparent", "translucent", "opaque"],
          "Pattern": ["plain", "striped", "floral", "painted(other than floral)"]
        },
        "Base": {
          "Shape": ["round", "rectangular", "square", "tripod", "ornate"],
          "Material": ["metal", "wood", "ceramic", "glass"],
          "Color": []
        },
        "Leg": {
          "Color": []
        }
      }
    },
    "LowProb": {
      "DirectAttributes": {
      },
      "Components": {
      }
    }
  }
}
```


Figure 15: Lamp attribute dictionary in JSON format.

Utensil


```
{
  "Utensil": {
    "HighProb": {
      "DirectAttributes": {
        "Type": ["spoon", "fork", "knife", "ladle", "spatula", "tongs", "bowl", "plate", "cup", "pan", "pot", "tray",
      "jar", "jug"]
      },
      "Components": {
        "OuterSurface": {
          "PrimaryColor": [],
          "Transparency": ["transparent", "translucent", "opaque"],
          "Pattern": ["plain", "horizontally striped", "vertically striped", "dotted", "floral", "geometric", "
     checked/chequered", "gradient"],
          "StripedColors": {"Stripe1Color": [], "Stripe2Color": []}
        },
        "InnerSurface": {
          "Color": []
        }
      }
    },
    "LowProb": {
      "DirectAttributes": {
      },
      "Components": {
        "Lid": {
          "Presence": ["yes", "no"],
          "Material": ["steel", "glass", "plastic", "ceramic"],
          "Color": [],
          "Shape": ["flat", "domed"]
        }
      }
    }
  }
}
```


Figure 16: Utensil attribute dictionary in JSON format.

You convert short natural-language scene descriptions into a structured scene graph.

• Objects (A_o) — the atoms: base noun phrases referring to entities (e.g., car, plant, table, person).
  - Represent objects by their lemma in lowercase (e.g., "persons" → "person").
  - Each object appears at most once in "objects", with its character-span [start, end) in the DESCRIPTION.

• Section 1 — Attribute/Action/State → Object (C_ao): inherent properties of a single atom
  - Use C_ao when a concept modifies exactly one object without introducing another object.
  - This includes: attributes (color, material, size, style), states (open/closed, broken/intact), and intransitive actions (walking, smiling, running) that describe the object itself.
  - Format: {"attr":"<string>","obj":"<object>","span":[start,end]} where attr is the phrase span from the DESCRIPTION that encodes the property or intransitive action.

• Section 2 — Object–Relation–Object (C_oro): concepts relating two atoms
  - Use C_oro for relations that link two objects, including transitive actions (holding, wearing, carrying, pushing), spatial/topological (left-of, under, behind, on, near), part/possession (with), etc.
  - Format: {"subj":"<object>","rel":"<string>","obj":"<object>","qualifiers":{...},"span":[start,end]}
  - Keep relation labels short and canonical (lowercase; hyphenate multiword), e.g., "closest-to", "sitting-on", "with", "over", "wearing", "holding-hand-of", "under", "left-of", "behind", "near".
  - If the text includes rank/ordinal (e.g., "second closest"), add {"rank":"second"} in "qualifiers".

• Plain objects
  - "plain_objects" are objects that do NOT appear as "obj" in any C_ao (attribute/state/intransitive action).
  - Being in a relation (C_oro) does NOT affect plainness.

General rules:
1) Use the DESCRIPTION string that follows to compute character spans [start, end) for every extracted item.
2) Lemmatize all object names (lowercase).
3) Prefer concise, normalized relation names; if unsure, use the closest canonical verb/preposition (e.g., "sitting-on", "closest-to", "with").
4) Do not hallucinate entities/attributes/relations; if truly ambiguous, skip.
5) Action disambiguation:
   - If an action does NOT require another object (intransitive), encode it as C_ao.
   - If an action targets another object (transitive), encode it as C_oro.

JSON SCHEMA (shape to follow):
```
{
 "objects": [{"name":"string","span":[int,int]}],
 "cao": [{"attr":"string","obj":"string","span":[int,int]}],
 "coro": [{"subj":"string","rel":"string","obj":"string","qualifiers":{}, "span":[int,int]}],
 "plain_objects": ["string"],
 "counts": {"relations": int, "cao": int, "plain_objects": int}
}
```

=== FEW-SHOT EXAMPLES ===

Example A (intransitive action as inherent property)
DESCRIPTION: person walking near the table

OUTPUT:
```
{
 "objects": [
 {"name": "person", "span": [0, 6]},
 {"name": "table", "span": [24, 29]}
 ],
 "cao": [
 {"attr": "walking", "obj": "person", "span": [7, 14]}
 ],
```

Figure 17: Pre-prompt for parsing referring expressions into attributes/actions, relations, and plain objects (1/3).

```
  "coro": [
    {"subj": "person", "rel": "near", "obj": "table", "qualifiers": {}, "span": [15, 29]}
  ],
  "plain_objects": ["table"],
  "counts": {"relations": 1, "cao": 1, "plain_objects": 1}
}

# Example 2 (ranked proximity + layered relations/attributes)
DESCRIPTION: person second closest to the table with a blue protective sheet over a white
cloth

OUTPUT:
{
  "objects": [
    {"name":"person","span":[0,6]},
    {"name":"table","span":[28,33]},
    {"name":"sheet","span":[49,54]},
    {"name":"cloth","span":[66,71]}
  ],
  "cao": [
    {"attr":"blue","obj":"sheet","span":[41,45]},
    {"attr":"protective","obj":"sheet","span":[46,56]},
    {"attr":"white","obj":"cloth","span":[61,66]}
  ],
  "coro": [
    {"subj":"person","rel":"closest-to","obj":"table","qualifiers":{"rank":"second"},"span":
[7,33]},
    {"subj":"table","rel":"with","obj":"sheet","qualifiers":{},"span":[34,54]},
    {"subj":"sheet","rel":"over","obj":"cloth","qualifiers":{},"span":[55,71]}
  ],
  "plain_objects": ["person","table"],
  "counts": {"relations":3,"cao":3,"plain_objects":2}
}

# Example 3 (transitive actions as relations)
DESCRIPTION: woman carrying a yellow bag under the flagpole, left of the man in a
black shirt

OUTPUT:
{
 "objects": [
 {"name":"woman","span":[0,5]},
 {"name":"bag","span":[20,23]},
 {"name":"flagpole","span":[34,42]},
 {"name":"man","span":[55,58]},
 {"name":"shirt","span":[69,74]}
 ],
 "cao": [
 {"attr":"yellow","obj":"bag","span":[17,23]},
 {"attr":"black","obj":"shirt","span":[66,74]}
 ],
 "coro": [
 {"subj":"woman","rel":"carrying","obj":"bag","qualifiers":{},"span":[6,23]},
 {"subj":"woman","rel":"under","obj":"flagpole","qualifiers":{},"span":[24,42]},
 {"subj":"woman","rel":"left-of","obj":"man","qualifiers":{},"span":[44,63]},
 {"subj":"man","rel":"wearing","obj":"shirt","qualifiers":{},"span":[59,74]}
 ],
 "plain_objects": ["woman","flagpole","man"],
 "counts": {"relations":4,"cao":2,"plain_objects":3}
}
```

(a) Pre-prompt for parsing referring expressions into attributes/actions, relations, and plain objects (2/3).

```
# Example 4
DESCRIPTION: person whose right hand is placed near their sunglasses or hair

OUTPUT:
{
 "objects": [
 {"name": "person", "span": [0, 6]},
 {"name": "hand", "span": [13, 17]},
 {"name": "sunglasses", "span": [42, 52]},
 {"name": "hair", "span": [56, 60]}
 ],
 "cao": [
 {"attr": "right", "obj": "hand", "span": [7, 12]},
 {"attr": "placed", "obj": "hand", "span": [18, 24]}
 ],
 "coro": [
 {"subj": "hand", "rel": "part-of", "obj": "person", "qualifiers": {}, "span": [7, 17]},
 {"subj": "person", "rel": "with", "obj": "sunglasses", "qualifiers": {}, "span": [37, 52]},
 {"subj": "person", "rel": "with", "obj": "hair", "qualifiers": {}, "span": [37, 60]},
 {"subj": "hand", "rel": "near", "obj": "sunglasses", "qualifiers": {}, "span": [25, 52]},
 {"subj": "hand", "rel": "near", "obj": "hair", "qualifiers": {}, "span": [25, 60]}
 ],
 "plain_objects": ["person", "sunglasses", "hair"],
 "counts": {"relations": 5, "cao": 2, "plain_objects": 3}
}
# Example 5
DESCRIPTION: man, holding the hand of the child wearing sleeveless jacket closest to
the person sitting on the red mattresses

OUTPUT:
{
 "objects": [
 {"name": "man", "span": [0, 3]},
 {"name": "hand", "span": [16, 20]},
 {"name": "child", "span": [28, 33]},
 {"name": "jacket", "span": [51, 57]},
 {"name": "person", "span": [73, 79]},
 {"name": "mattress", "span": [94, 104]}
 ],
 "cao": [
 {"attr": "sleeveless", "obj": "jacket", "span": [41, 51]},
 {"attr": "sitting", "obj": "person", "span": [80, 87]},
 {"attr": "red", "obj": "mattress", "span": [90, 93]}
 ],
 "coro": [
 {"subj": "man", "rel": "holding", "obj": "hand", "qualifiers": {}, "span": [5, 20]},
 {"subj": "hand", "rel": "part-of", "obj": "child", "qualifiers": {}, "span": [21, 33]},
 {"subj": "child", "rel": "wearing", "obj": "jacket", "qualifiers": {}, "span": [34, 57]},
 {"subj": "child", "rel": "closest-to", "obj": "person", "qualifiers": {"rank": "first"},
"span": [58, 79]},
 {"subj": "person", "rel": "sitting-on", "obj": "mattress", "qualifiers": {}, "span": [80,
104]}
 ],
 "plain_objects": ["man", "hand", "child"],
 "counts": {"relations": 5, "cao": 3, "plain_objects": 3}
}

=== END OF EXAMPLES ===
```

(b) Pre-prompt for parsing referring expressions into attributes/actions, relations, and plain objects (3/3).

A.6 Human Answering and Verification Protocol

Human Answering

For the first round of human answering, we implemented the following standardized protocol to ensure consistency across annotators:

- Make sure the referring expression correctly localizes the target object. Annotators should first read the referring expression and identify which object it localizes. Once identified, they can toggle the bounding box to verify that the localized object exactly matches the content of the bounding box. If the expression is ambiguous or does not match the annotated object, the annotator should flag the expression. As shown in Fig. 19, annotators can mark descriptions that do not correspond to the object they initially interpreted. These ambiguous expressions are then refined and sent for a second round of human verification.
- Once the referring expressions are verified, annotators proceed to answer the questions.
- While answering, annotators must be confident about their responses; if they are unsure, they can mark the question as *unsure* (as shown in Fig. 19e).
- If the annotators feel that a question is invalid for a particular object, they can mark the question invalid (as shown in Fig. 19c), or leave it unanswered.
- Any component or attribute of an object for which no physically visible answer is present, then it should be marked as unanswerable (as shown in Fig. 19f) or left unanswered.
- If the answer to the question is not included in the options, the annotator is requested to suggest the answer by clicking on 'None of the Above'.
- If any question inadvertently includes the answer itself, human annotators are requested to mark the question as invalid. Such questions are discarded from the benchmark.
- For numerical counting, annotators are required to report the number of target instances corresponding to each query within a group. They must also verify that the sum of the counts for all child queries matches the count provided for the parent query. If this consistency check fails, the entire group is flagged and returned to the question annotators for refinement.

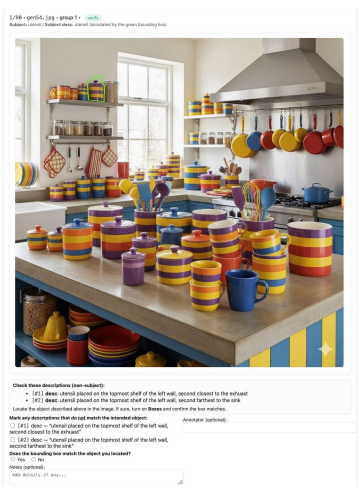
Human Verification

All answered questions and refined expressions from the human answering stage are forwarded for verification.

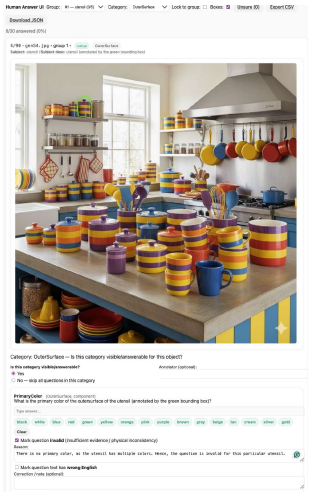
- In this stage, human verifiers confirm that each referring expression uniquely and unambiguously identifies the intended target object. They also ensure that the answer selected in the previous stage objectively and correctly answers the queries. If both conditions are satisfied, the question–answer pair is accepted into the benchmark.
- When an annotator from the answering stage selected 'None of the Above' and provided an alternative answer, the verifier must assess whether this suggestion was necessary and whether it is objectively correct. If the verifier agrees with the suggestion, the question is retained by appending the proposed answer to the option list. The attribute dictionaries are updated accordingly to incorporate this new answer.
- For numerical counting tasks, annotators check the numerical estimates associated with each query within a group. They additionally confirm that the sum of all child-query counts matches the parent-query count, ensuring hierarchical consistency.
- For numerical counting, the groups that were revised during the first stage, a joint review by two independent annotators is conducted. This step ensures that all refinements are accurate and that the numerical annotations are fully consistent before inclusion in the benchmark.



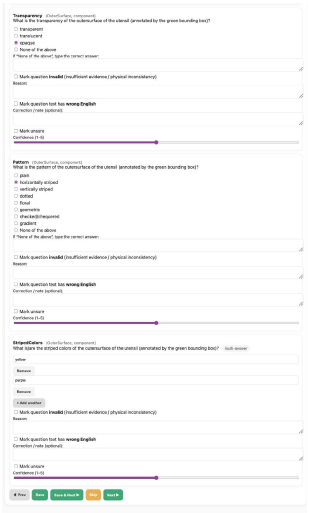
(a) Description without bounding box.



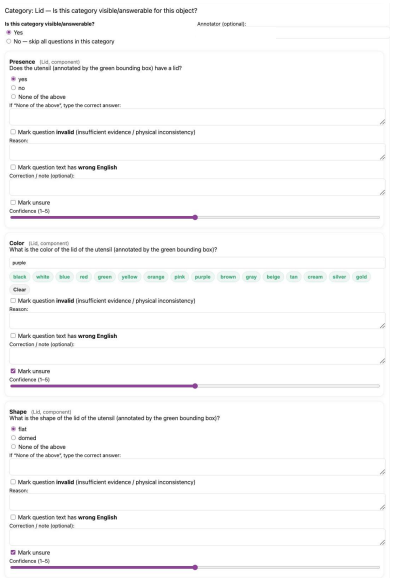
(b) Description with bounding box.



(c) Invalid question example.



(d) Valid question example.



(e) Ambiguous question example.



(f) Non-answerable attribute example.

Figure 19: Human annotation interface examples used for referring-expression verification and QA filtering. Panels show target verification with and without bounding boxes, invalid and valid question cases, ambiguous questions, and non-answerable attributes.

The entire human annotation process was done by 9 annotators. It was ensured that annotators only verified the questions answered by other annotators.

Human Baseline. The human baseline reported in Table 2 was performed by six individuals who were not involved in the annotation of MIMO-Bench. Each solved 25 groups for attribute binding and 15 for numerical counting.

Refer to Fig. 20 to see how the attribute binding question for a synthetic image gets narrowed down to a final set of questions. MLLMs are asked the final set of questions for each of the referring expressions. Fig 21 shows the formation of the numerical counting questions for a synthetic image.

A.7 Scalability of Benchmark Construction

MIMO-Bench is designed as a quality-controlled benchmark rather than a fully automatic data-generation pipeline. The most expensive stages are human-dependent: annotators must create or verify referring ex-

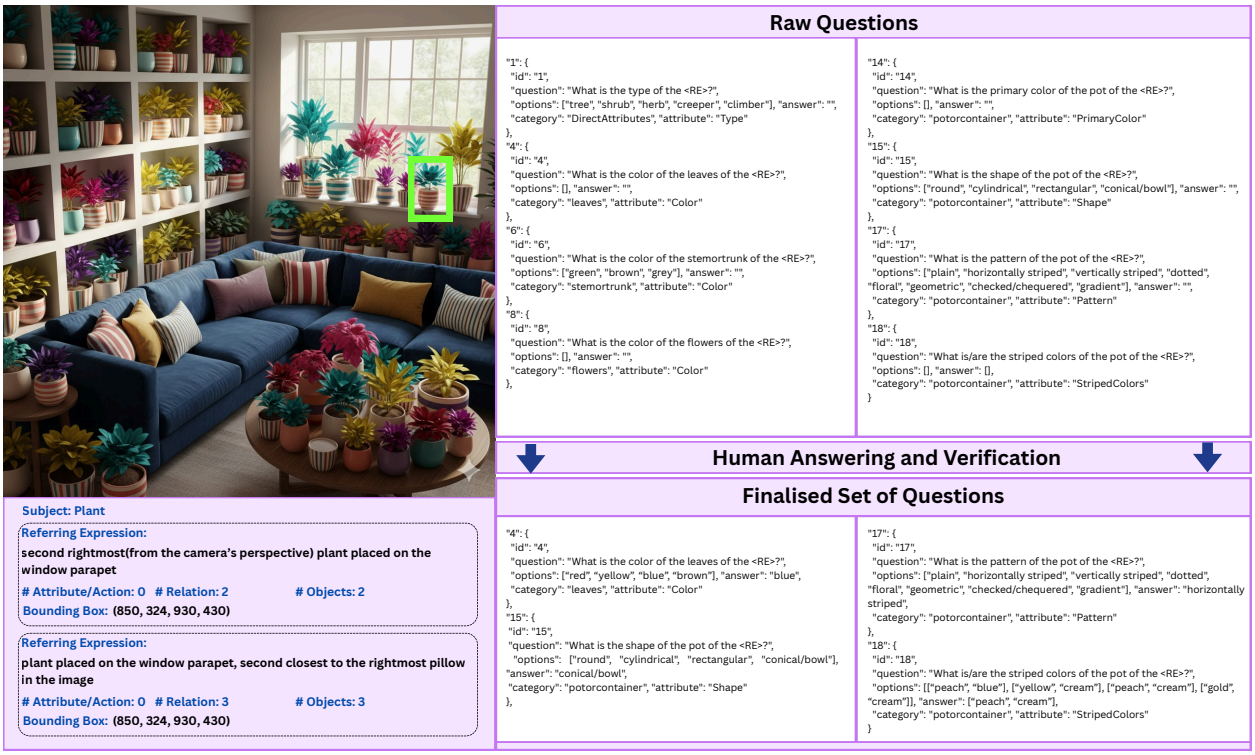


Figure 20: Sample question dictionary of attribute binding getting modified from raw question answers to the final set of questions, having gone through human answering and verification.

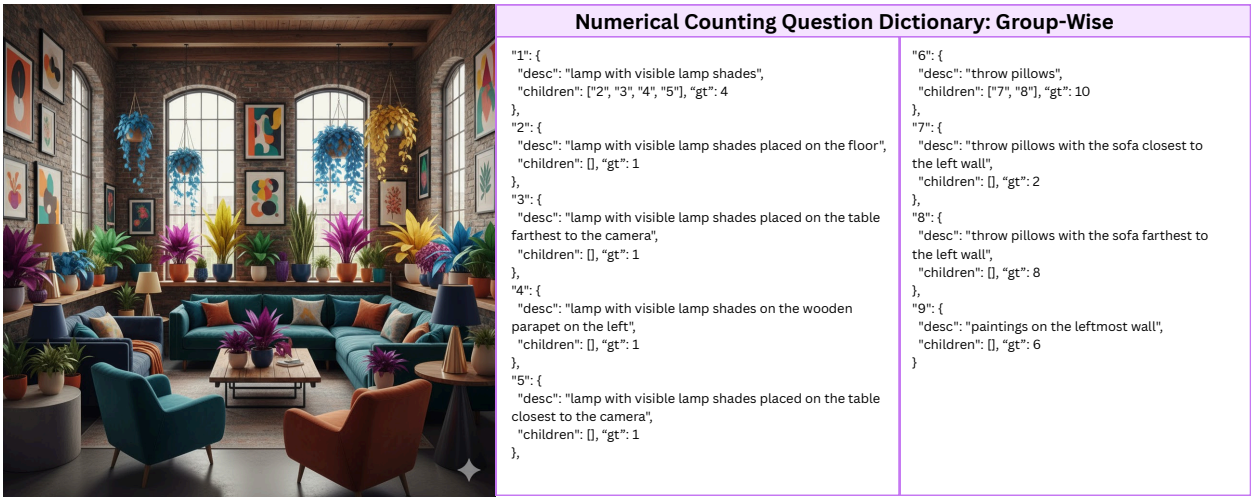
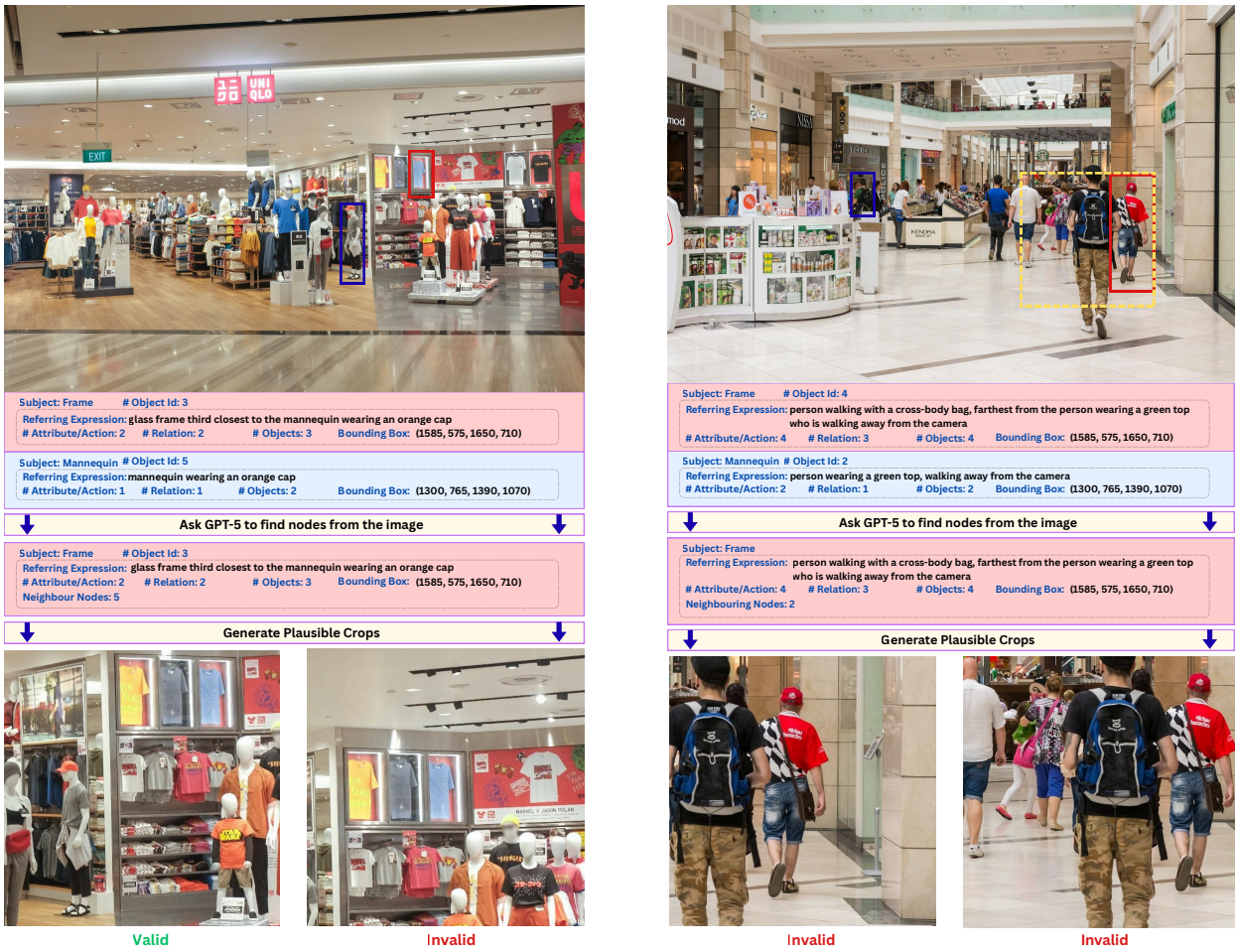


Figure 21: Sample question dictionary for numerical counting task. A parent node will have a non-empty "children" list as seen for nodes 1 and 6.

pressions, ensure that each expression uniquely localizes the intended target, discard ambiguous or non-answerable questions, and verify the final answers. This makes scaling MIMO-Bench challenging compared to benchmarks that rely only on templated generation or automatically mined annotations.

At the same time, the pipeline is partially scalable. Once images, object masks, and referring expressions are available, the attribute dictionary and template-based question generator automatically produce a large pool of candidate attribute-binding questions, while numerical-counting questions are generated from group-level



(a) Feasible crop.

(b) Rejected crop.

Figure 22: MIMO-Crop examples. Left: a feasible crop where the target and referenced neighboring object are both retained. Right: a rejected case where no crop can preserve all objects mentioned in the referring expression.

annotations. Human effort is therefore concentrated on filtering and verification rather than writing every question from scratch. This design is intentional: in the MIMO setting, small ambiguity in the referring expression or target object can change the answer, so fully automated scaling would likely introduce noisy QA pairs. We therefore prioritize visually grounded and unambiguous examples over maximizing dataset size. Future extensions can scale the benchmark by improving automatic referring-expression proposal and ambiguity detection, while retaining human verification as the final quality-control step.

B MIMO-Crop

We curate MIMO-Crop using a semi-automated pipeline. For each annotated object in an image of MIMO-Bench, we already have a referring expression and a bounding polygon annotation. We want to crop the images from MIMO-Bench in such a manner that the cropped version still uses the same referring expressions. To achieve this, we must ensure that every referenced object in the expression is present in the crop. Otherwise, the crop becomes invalid, as the expression would refer to neighboring objects no longer visible. Therefore, we create a cropped image corresponding to each expression of each annotated object in every image.

For each annotated object in an image, we use GPT-5 to identify how many other nodes (objects) are mentioned in their expression, shown in Fig. 22a. If the target object has annotated neighboring nodes in its expression (Figs. 22a and 22b), our cropping script aims to create a crop that can contain these neighboring nodes in the crop. As seen in Fig. 22a, some object expressions have neighboring nodes' bounding boxes close enough, hence our cropping script automatically includes them in the plausible crops. After generating plausible crops, we send them for human verification. In cases like Fig. 22b, no crop can contain the neighboring object referred to in the expression. Consequently, such object-expression pairs are automatically rejected from MIMO-Crop.

In the above cases, GPT-5 (OpenAI, 2025b) can correctly parse neighboring nodes as corresponding to a single coherent annotated object. However, in some instances, the neighboring nodes do not correspond to a single annotated object, but describe regions or composite structures ("balcony", "window parapet", and "street"). For these object–expression pairs, we randomly generate candidate crops around the target object and manually verify whether the referenced components are actually present in the image. If the mentioned nodes cannot be visually confirmed, the corresponding samples are discarded. Throughout the whole process, approximately 7% of the object-expression pairs were discarded, suggesting that generating random crops for the majority suffices to create a plausible crop.

C Additional Results

C.1 Evaluation Setup

To ensure consistency across evaluation runs, we use greedy decoding for all models unless specified otherwise. Models such as GPT-5 (OpenAI, 2025b) (minimal and medium-effort reasoning modes), G3.5-Flash (Google DeepMind, 2026b) and G3.1-Pro-Preview (Google DeepMind, 2026a) (low reasoning effort), G2.5 (Pro and Flash) (Comanici et al., 2025), GLM-4.5V (GLM-V Team et al., 2025), IVL3.5-241B-A28B (Wang et al., 2025a), and IVL3-78B (Zhu et al., 2025) are accessed through their respective APIs. Locally evaluated open-weight models, including GLM-4.1V-9B-Thinking, are run from their publicly available HuggingFace weights. Local experiments are conducted on a mix of hardware: a local cluster equipped with two NVIDIA L40S GPUs (48 GB VRAM each) and RunPod instances with an NVIDIA A100 PCIe GPU (80 GB VRAM).

We strictly prompt the models to provide answers within <answer> </answer> XML tags. The exact prompts used for attribute binding and numerical counting are shown in Figs. 23 and 24, respectively.

C.2 Exact Group Accuracy

The reported group accuracy G_{ACC} is group-balanced rather than all-or-nothing: it averages each target's fraction of correct valid-attribute answers with equal weight per target. To complement it, Exact Group Accuracy counts a target as correct only if every valid attribute question is correct. The word "group" names the aggregation unit rather than a strict success criterion: attributes belonging to one target are scored together, and each target group receives equal weight regardless of its number of valid attributes. This prevents richly annotated targets from dominating and is why we retain the name G_{ACC} while clarifying the surrounding prose. Table 8 reports the strict Exact Group Accuracy alongside G_{ACC}.

Exact Group Accuracy ranges from 1.15% to 33.33%, confirming that complete all-attribute success is substantially rarer than partial within-group accuracy. The two metrics therefore answer complementary questions.

C.3 Parent-Level Global–Local Counting Comparison

Table 9 shows that several models benefit more from localized decomposition than from full-scene counting at the parent-group level. This is visible both in the negative sign(Δ_{GL}) entries[5] and in the frequency columns: for example, IVL3-8B favors local counting on 61.2% of parent groups versus 28.6% global, and Q3-8B-

[5]For some models the gap on the real set can be larger than on the synthetic set; however, the synthetic set accounts for a larger volume of the numerical-counting parent groups and hence dominates the aggregate direction.

```
You are a precise visual question answering assistant.

Answer the question about the provided image.

Question (use EXACTLY as written below; do NOT rewrite or reformat):
{q}

RESPONSE FORMAT (STRICT):
- Write at most one short sentence (optional).
- Then append EXACTLY ONE XML tag on the SAME line at the very end: <answer>SHORT_ANSWER</answer>
- Do NOT add anything after </answer>. Do NOT add extra tags, code fences, or headings.

SHORT_ANSWER RULES:
- If the question text contains labeled options on separate lines (e.g., "A. ...", "B) ...", "1) ..."):
  - Choose the correct option label only (A, B, 1, 2, ...).
  - Output exactly that label inside the tag, e.g., <answer>A</answer>.
- Otherwise (no labeled options in the text):
  - Use a minimal span (e.g., "green", "yes", "no", "3", or an object name).
  - lowercase, no punctuation.
  - If unsure or not visible, use "unknown".

EXAMPLES (follow format precisely):
- The cup appears light green. <answer>green</answer>
- The observation matches choice B. <answer>B</answer>
- Not visible. <answer>unknown</answer>

Now answer:
```

Figure 23: Prompt used for attribute-binding questions.

```
You are a precise visual counting assistant.

Answer the question about the provided image.

Question (use EXACTLY as written below; do NOT rewrite or reformat):
Count the number of {desc_text}

RESPONSE FORMAT (STRICT):
- Write at most one short sentence (optional).
- Then append EXACTLY ONE XML tag on the SAME line at the very end: <answer>N</answer>
- Do NOT add anything after </answer>. Do NOT add extra tags, code fences, or headings.

COUNTING RULES:
- Count visible instances only; if ambiguous or heavily occluded, count clearly visible ones.
- If none are visible, answer 0.
- Output just the integer N inside <answer>.

EXAMPLES (follow format precisely):
- There are three. <answer>3</answer>
- None present. <answer>0</answer>

Now answer:
```

Figure 24: Prompt used for numerical-counting questions.

Table 8: Exact Group Accuracy (%): the percentage of target groups for which all valid attribute questions are answered correctly.

Model	Exact group (%)	Model	Exact group (%)
Gemini-2.5-Flash	33.33%	InternVL3.5-8B	3.92%
Qwen3-VL-8B-Thinking	15.83%	InternVL3.5-38B	3.61%
InternVL3.5-241B-A28B	11.75%	InternVL3-8B	3.55%
Qwen3-VL-32B-Thinking	9.78%	InternVL3-14B	3.41%
Gemini-2.5-Pro	9.39%	Qwen3-VL-8B-Instruct	3.13%
Qwen2.5-VL-72B-Instruct	6.94%	Qwen3-VL-30B-A3B-Thinking	2.70%
Qwen3-VL-32B-Instruct	6.14%	Qwen2.5-VL-7B-Instruct	2.34%
InternVL3-78B	5.43%	Qwen3-VL-30B-A3B-Instruct	2.08%
GLM-4.1V-9B-Thinking	5.42%	InternVL3.5-14B	1.71%
InternVL3-38B	5.17%	GLM-4.5V	1.15%
Qwen2.5-VL-32B-Instruct	4.60%		

Table 9: Parent-level comparison between global and localized counting on the parent groups. "G>L" and "L>G" give the percentage of parent groups where the global score is strictly greater or strictly smaller than the localized child average, respectively. $\mathrm{sign}(\Delta_{GL})$ reports the direction of the gap $\mathcal{G}_{\mathrm{MRA}} - \mathcal{L}_{\mathrm{MRA}}$ over the real and synthetic sets. $\sim$ denotes a near-tie. A negative sign together with L>G % exceeding G>L % indicates that localized decomposition helps.

Model	**sign(Δ_{GL})**	**G>L (%)**	**L>G (%)**
G2.5-Flash	+	42.1	43.9
G2.5-Pro	+	61.4	29.8
GPT-5-medium	+	59.6	31.6
Q3-32B-Instruct	−	31.6	54.4
Q3-8B-Instruct	−	33.3	52.6
Q3-30B-A3B-Instruct	−	29.8	56.1
Q2.5-72B-Instruct	−	40.4	49.1
Q2.5-32B-Instruct	+	49.1	31.6
Q2.5-7B-Instruct	−	28.1	47.4
IVL3.5-241B-A28B	−	0.0	17.5
IVL3.5-38B	−	38.6	45.6
IVL3.5-14B	+	47.4	42.1
IVL3.5-8B	+	47.4	47.4
IVL3-78B	$\sim$	38.6	38.6
IVL3-38B	−	35.7	50.0
IVL3-14B	+	41.7	50.0
IVL3-8B	−	28.6	61.2
GLM-4.5V	−	40.4	49.1
GLM-4.1V-9B-Thinking	−	0.0	66.7

Instruct on 52.6% versus 33.3%. By contrast, G2.5-Pro and GPT-5-medium favor global counting more often than local counting. These parent-level comparisons confirm that the main-text differences between $\mathcal{G}_{\mathrm{MRA}}$ and $\mathcal{L}_{\mathrm{MRA}}$ are not solely artifacts of averaging across groups.

For numerical counting, we additionally test internal hierarchy preservation. For parent–children group g, let $e_g = \hat{y}^{(g)}_{\mathrm{parent}} - \sum_{c \in \mathcal{C}(g)} \hat{y}^{(g)}_{c}$. Exact consistency is the percentage of groups with $e_g = 0$; MAE is the mean $|e_g|$; signed bias is the mean e_g. These metrics measure internal consistency, not correctness against ground truth. Table 10 reports these hierarchical-consistency measures.

Consistency varies from 3.8% to 72.7%, and signed bias is negative for every model. Thus, when predictions disagree, the parent estimate is generally below the sum of localized child estimates. Consistency should not be interpreted as accuracy: mutually consistent predictions can still differ from ground truth.

Table 10: Hierarchical consistency of numerical-counting predictions. Negative bias means the predicted parent count is smaller than the sum of predicted child counts.

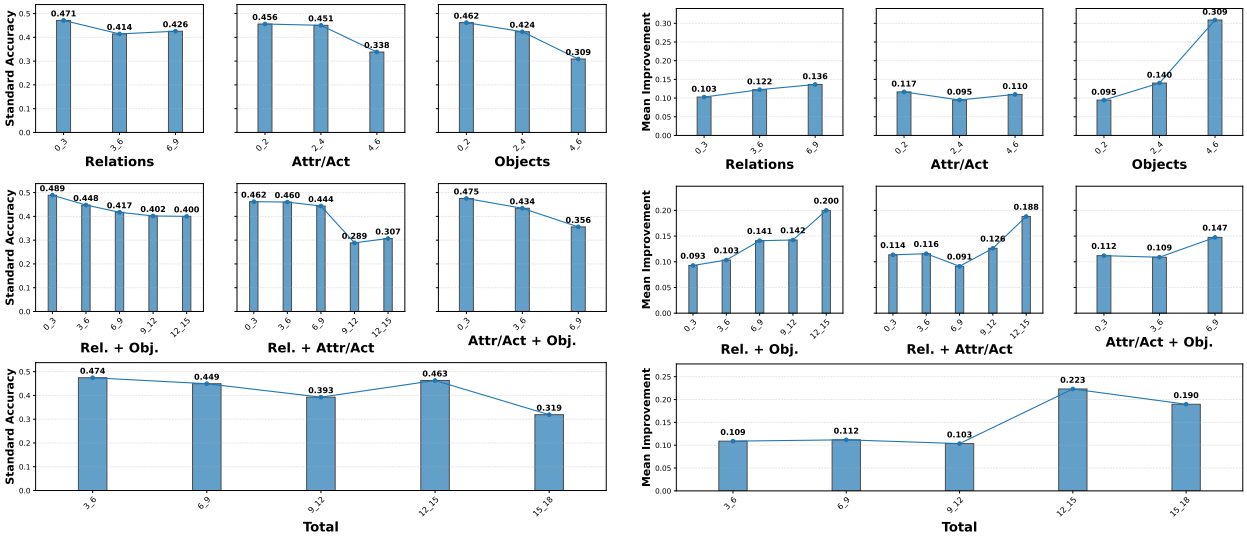
Model	Exact Cons. (%)	MAE	Bias
IVL3.5-241B-A28B	72.7	1.00	-1.00
Q3-32B-Thinking	50.0	0.75	-0.35
G3.1-Pro-Preview-Low	45.3	1.55	-0.19
G3.5-Flash-Low	45.3	1.34	-0.36
G3.5-Flash-High	43.4	1.58	-1.02
Q3-8B-Thinking	42.9	0.86	-0.86
GLM-4.1V-9B-Thinking	25.0	2.00	-0.75
Q2.5-72B	22.6	3.42	-1.57
IVL3-78B	20.8	3.62	-3.36
GPT-5-medium	19.2	2.37	-0.90
GLM-4.5V	18.9	3.17	-1.70
Q3-32B-Instruct	18.9	2.87	-2.34
G2.5-Pro	17.0	2.74	-1.45
Q3-30B-A3B-Instruct	17.0	4.04	-3.55
G3.1-Pro-Preview-High	15.2	1.55	-0.39
Q2.5-32B	11.3	3.53	-2.70
Q2.5-7B	11.3	3.68	-2.81
Q3-8B-Instruct	11.3	3.45	-2.96
G2.5-Flash	10.2	4.84	-2.76
IVL3-8B	9.8	6.20	-6.20
IVL3-14B	9.1	5.55	-4.15
IVL3-38B	7.7	3.71	-3.25
IVL3.5-38B	7.5	4.42	-4.00
IVL3.5-14B	3.8	4.72	-4.42
IVL3.5-8B	3.8	3.98	-3.79

C.4 Full Count-Conditioned Numerical Counting Statistics

Table 11 makes two trends explicit. First, low-count bins are prone to false inclusion: at GT 0, exact match is only 0.3333 and the average bias is positive (+1.0833). Second, as GT count increases, the dominant failure mode shifts toward undercounting: the bias becomes negative starting at GT 4 and becomes substantially more negative in high-count bins, while off-by-two-or-more errors dominate. This extends the main-text discussion with the full per-GT error profile while keeping the presentation compact.

Table 11: Full count-conditioned numerical-counting statistics aggregated across all available model predictions. Rows are grouped by ground-truth (GT) target-set size. We report mean $\mathcal{MRA}$, exact-match rate, signed bias ($\hat{y} - y$), mean absolute error (MAE), off-by-one rate ($|\hat{y} - y| \leq 1$), and off-by-two-or-more rate ($|\hat{y} - y| \geq 2$). Positive bias indicates overcounting and negative bias indicates undercounting. Per-GT sample frequencies are intentionally omitted for compactness.

GT	$\mathcal{MRA}$	**Exact**	**Bias**	**MAE**	**Off-by-1**	**Off-by-2+**
0	0.3333	0.3333	+1.0833	1.0833	0.6111	0.3889
1	0.4069	0.4069	+0.7430	0.9272	0.8223	0.1777
2	0.3555	0.3555	+0.4947	1.1089	0.6838	0.3162
3	0.4968	0.3541	+0.5889	1.4324	0.7109	0.2891
4	0.4263	0.2564	-0.5406	1.4124	0.5962	0.4038
5	0.5234	0.2888	-0.6231	1.6261	0.6231	0.3769
6	0.4261	0.2012	-1.5045	2.2913	0.4384	0.5616
7	0.3270	0.0292	-2.2774	3.0949	0.2299	0.7701
8	0.3976	0.0714	-2.0952	3.1905	0.2619	0.7381
9	0.4583	0.0709	-2.0000	2.9764	0.2756	0.7244
10	0.3734	0.0781	-3.1172	3.6953	0.1719	0.8281
11	0.3294	0.0196	-4.5490	4.8235	0.2549	0.7451
12	0.3882	0.1319	-3.8889	4.3194	0.1736	0.8264
13	0.4941	0.0588	-3.5294	3.8824	0.1765	0.8235
14	0.0500	0.0000	-9.5556	9.6667	0.0556	0.9444
16	0.4083	0.0000	-5.5278	5.6389	0.0556	0.9444
17	0.1750	0.0278	-8.5556	11.2222	0.0278	0.9722
19	0.3611	0.0000	-6.0000	7.0000	0.1111	0.8889



(a) Standard accuracy.

(b) Absolute gain with visual prompting.

Figure 25: Compact view of how referring-expression composition relates to standard accuracy and absolute improvement under visual prompting.

Most numerical-counting questions have ground-truth counts between 1 and 12. Results for individual bins above 12 should therefore be treated as unstable; aggregate conclusions are based primarily on the well-supported 1–12 range.

C.5 Visual Prompting Discussion

Method	512 × 512	1024 × 1024	MIMO-Bench	WVP
	$\mathcal{I}_{\text{ACC}}$	$\mathcal{I}_{\text{ACC}}$	$\mathcal{I}_{\text{ACC}}$	$\mathcal{I}_{\text{ACC}}$
Evaluated MLLMs				
Qwen3-VL				
32B	65.5	56.8	46.3	64.3
30B-A3B	52.9	45.2	35.2	59.2
8B	56.1	50.6	39.2	64.7
Qwen2.5-VL				
32B	55.6	52.7	40.6	63.1
InternVL3.5				
38B	57.3	49.3	41.7	54.1
14B	42.5	37.0	35.4	46.6
8B	47.1	41.5	39.2	45.2
InternVL3				
78B	49.3	46.9	40.5	–[†]
38B	54.1	50.7	45.5	56.3
14B	53.2	48.3	36.1	49.3
8B	44.0	38.8	37.3	45.9
GLM-4.5V	61.7	55.1	46.2	64.7

Table 12: Comparison of MLLM performance on the subset used for visual-prompting and graying ablations. The 512 × 512, 1024 × 1024, and MIMO-Bench columns correspond to the graying setup with progressively more visible distractors, while WVP reports performance with visual prompting. [†] We could not obtain a WVP result for IVL3-78B because of API server issues.

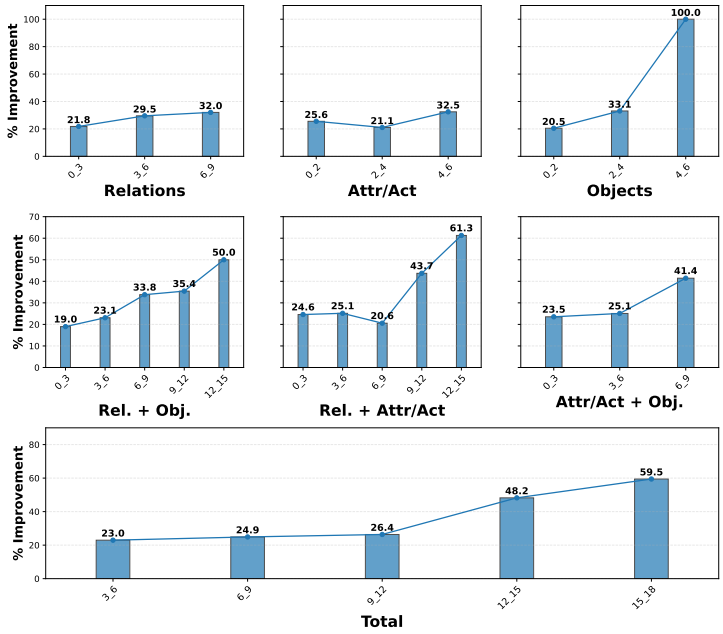


Figure 26: Percentage gain with visual prompting.

To recap, visual prompting in attribute binding helps us in removing the need of referring expressions for isolation of the target object, by replacing them with a visual marker/prompt over the target. Referring expressions, as discussed in Sec. 3.2 of the main paper, pose multiple subtasks to the model in the quest to localize the target object. Visual prompting essentially reduces the number of visual subtasks posed to MLLMs to 2 (object recognition and bounding box recognition). Kindly refer to Fig. 26 to visualize how performance increases as the referring expression gets shorter and a visual marker is used.

Figs. 25a, 25b, and 26 show how questions with varying compositions (attributes/actions, relations, and objects) of referring expressions behave in different settings. The first row in each figure reports the corresponding performance as the count (binned) of a single component in the expression increases in isolation. The second row reports performance for pairwise combinations of components, whereas the third row reports performance for expressions involving all three components together. The y-axis in all plots shows performance averaged across all models presented in Fig. 5 of the main paper.

Fig. 25a displays how $\mathcal{I}_{\text{ACC}}$ in the standard setting (without visual prompting) varies with the composition of the expression. The results suggest that as the count of the components, such as attributes/actions, relations, and objects, increases, either in isolation or in combination, the accuracy in standard evaluation largely decreases monotonically. The plot essentially tells us that increasing any of the components in the expression leads to a decrease in the average standard accuracy of models, thereby empirically demonstrating the complexity induced by the components of the referring expression. Specifically, we notice that the standard accuracy drops steeply when more than four attributes/actions and objects are present in the expression. In contrast, the standard accuracy does not degrade as steeply when the number of relations in an expression increases.

Figs. 25b and 26 show the mean absolute and percentage improvement in $\mathcal{I}_{\text{ACC}}$ provided by visual prompting over the standard setting, as a function of the composition of the referring expressions. The main trend suggests that the larger the count of the three components in the expressions, the larger the gains provided by visual prompting. Taken together with the findings from Fig. 25a, this highlights how the volume of subtasks posed by these components, whether in isolation or in combination, substantially hinders MLLMs' performance, and how providing relief via visual prompting is especially beneficial for the most complex expressions. In particular, expressions with higher component counts exhibit both lower standard accuracy and larger improvements with visual prompting, indicating that referring expressions in the MIMO setting are a key bottleneck for current models.

C.5.1 Grounding and Localization

We directly evaluate referring-expression (RE) grounding on the same examples used for attribute binding (AB). Box-choice grounding asks the model to select among four marked candidates (25% chance), whereas coordinate grounding leaves the image unchanged and requires a bounding-box output in normalized 0–1000 coordinates (approximately 9% chance under a uniform choice among annotated instances). Each interface is run jointly with the AB question and separately in isolation. A coordinate prediction is correct when the annotated instance with maximum IoU is the target. The four joint cells and conditional probabilities come from the joint run; isolated grounding accuracy comes from the separate grounding-only run and is therefore not the sum of the joint G^+A^+ and G^+A^- cells. G^- includes both failures to return a usable location and selections of the wrong instance. Table 13 reports the joint grounding and answer outcomes, and Table 14 the isolated grounding accuracy together with AB accuracy conditioned on grounding.

Table 13: Explicit RE-grounding outcomes and their correspondence with AB answers on matched examples. Cells are percentages. G^+/G^- denote correct/incorrect grounding and A^+/A^- correct/incorrect AB answers.

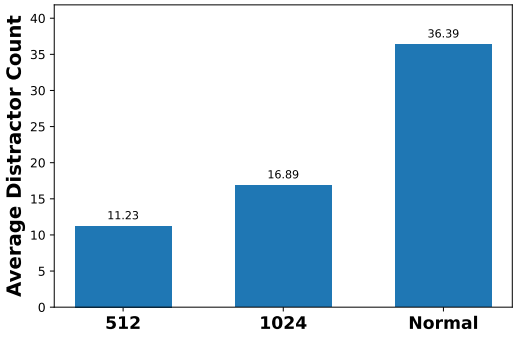
Model	Grounding	G^+A^+	G^+A^-	G^-A^+	G^-A^-
Qwen3-VL-8B	Box-choice	32.6	35.8	12.9	18.7
	Coordinates	14.5	9.9	33.3	42.2
Qwen2.5-VL-7B	Box-choice	19.7	36.8	13.7	29.8
	Coordinates	2.6	3.2	34.5	59.7
InternVL3-8B	Box-choice	23.2	27.6	21.6	27.6
	Coordinates	2.5	2.2	45.8	49.4
InternVL3.5-8B	Box-choice	18.9	33.5	15.6	32.0
	Coordinates	6.0	8.6	30.2	55.3
InternVL3-14B	Box-choice	26.5	23.2	21.3	29.1
	Coordinates	6.3	4.4	49.6	39.7
InternVL3.5-14B	Box-choice	24.5	28.5	13.7	33.3
	Coordinates	5.5	5.7	35.9	52.9

Table 14: Grounding-only accuracy and conditional AB accuracy on the same matched examples. Isolated G-Acc. comes from a separate grounding-only run. $\Delta = P(A^+ \mid G^+) - P(A^+ \mid G^-)$. Joint box-choice AB values use an overlaid image and are not directly comparable with the main benchmark $\mathcal{I}_{\mathrm{ACC}}$ values.

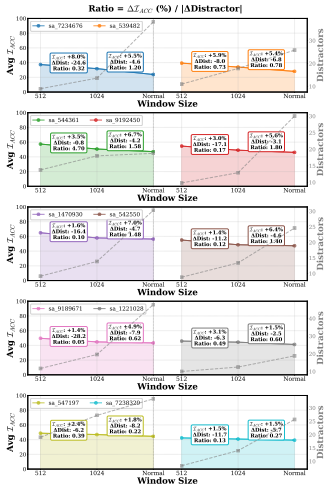
Model	Grounding	Isolated G-Acc.	$P(A^+ \mid G^+)$	$P(A^+ \mid G^-)$	Δ
Qwen3-VL-8B	Box-choice	59.3	47.6	40.8	+6.9
	Coordinates	25.1	59.4	44.1	+15.3
Qwen2.5-VL-7B	Box-choice	53.8	34.8	31.4	+3.4
	Coordinates	8.3	44.5	36.6	+7.9
InternVL3-8B	Box-choice	52.9	45.7	44.0	+1.7
	Coordinates	9.1	53.2	48.1	+5.1
InternVL3.5-8B	Box-choice	56.4	36.1	32.8	+3.2
	Coordinates	17.5	41.0	35.3	+5.7
InternVL3-14B	Box-choice	54.4	53.3	42.3	+11.0
	Coordinates	15.3	58.7	55.5	+3.2
InternVL3.5-14B	Box-choice	54.8	46.3	29.1	+17.1
	Coordinates	17.9	48.8	40.5	+8.4

Isolated box-choice grounding reaches 52.9–59.3%, while isolated coordinate grounding reaches 8.3–25.1%. In the joint runs, correct grounding is associated with a 1.7–17.1 point improvement in AB accuracy, but $P(A^+ \mid G^+)$ remains only 34.8–53.3%. Thus, grounding failures and target-conditioned downstream errors both remain; the association is not interpreted as a causal decomposition because grounding difficulty can correlate with answering difficulty and each interface introduces its own requirements.

The two interfaces serve different analytical roles. The isolated coordinate rate exceeds the corresponding joint coordinate rate for all six models by 0.7–6.7 points, so the format gap is not created by the joint prompt. Relative to chance, isolated box choice achieves 2.1–2.4 times its 25% chance level, whereas isolated coordinates achieve 0.9–2.8 times the approximately 9.0% chance level; Qwen2.5-VL-7B and InternVL3-8B are at chance under coordinates. The coordinate chance level corresponds to a uniform choice among the annotated instances in an image, whose median count is approximately 14. Much of the raw accuracy gap between the two interfaces therefore reflects the answer space shrinking from roughly 14 annotated



(a) Average distractor count across window sizes.



(b) Image-wise relationship between distractor reduction and $\mathcal{I}_{\mathrm{ACC}}$ gain.

Figure 27: Graying experiment diagnostics. The first panel summarizes average distractor count; the second shows that the accuracy gain per removed distractor varies substantially across images and window sizes.

instances to four marked candidates, rather than a corresponding change in underlying localization ability. We therefore use isolated coordinates as the less intrusive localization readout, while using joint box choice for the conditional analysis: correctly grounded cases comprise 49.7–68.4% of the box-choice examples but only 4.7–24.4% of the coordinate examples, giving a substantially more stable estimate of $P(A^+ \mid G^+)$. This elicitation dependence is also why we did not make explicit localization a mandatory intermediate benchmark stage. The released benchmark materials will include the target masks and bounding-box coordinates so that alternative grounding interfaces can be evaluated. The annotations already contain bounding-polygon coordinates for every target object, allowing developers to evaluate either overlaid bounding boxes or raw coordinate outputs.

C.6 Graying and Distractor-Reduction Controls

We first quantify how the graying intervention changes the number of visible same-category distractors. We use SAM3 (Carion et al., 2026), a promptable concept segmentation model, to count instances of the target object category within each retained window. For each window, we supply the object-category name as a text concept prompt, and SAM3 returns one instance-level segmentation mask per detected instance. We use the number of returned masks as the distractor count. Fig. 27a reports the resulting average distractor count for each window size.

Fig. 27b relates this change in distractor burden to model performance for IVL3-8B, 14B, and 38B; IVL3.5-8B, 14B, and 38B; and Q3-8B-Instruct and 32B-Instruct over a set of 10 images. For each window size $w \in \{512, 1024, \mathrm{Normal}\}$, we denote the average $\mathcal{I}_{\mathrm{ACC}}$ across models for image i by $I_i(w)$ and the corresponding distractor count by $D_i(w)$.

The callouts in Fig. 27b summarize the change relative to the unmodified image:

$$\Delta I_i(w) = I_i(w) - I_i(\mathrm{Normal}), \tag{5}$$

$$\Delta D_i(w) = D_i(w) - D_i(\mathrm{Normal}), \tag{6}$$

$$R_i(w) = \frac{\Delta I_i(w)}{|\Delta D_i(w)|}, \tag{7}$$

where $R_i(w)$ can be interpreted as the accuracy gain per distractor removed.

The trajectories of $I_i(w)$ and $D_i(w)$ are not aligned across images or window sizes. Although accuracy generally improves as fewer distractors remain visually active, $R_i(w)$ varies substantially across images and between the 512 and 1024 conditions for the same image. Thus, there is no single global slope relating distractor count to accuracy, and distractor count alone does not predict the magnitude of the performance change. Other properties, such as the spatial configuration of distractors, may also affect the relationship between distractor reduction and model performance.

Importantly, graying is not a clean intervention on distractor count alone. Suppressing the surrounding region also removes contextual information and increases the relative salience of the retained region. We therefore interpret graying as an intervention that reduces *effective visual distractor burden*, rather than as a causal manipulation of distractor count in isolation.

Reducing distractors in a natural image necessarily operates through some change to the visual input. Cropping removes surrounding content while also changing target scale and effective resolution. Graying and blurring preserve the image extent and, for a given model, the visual-token count, but reduce visual competition by suppressing the surrounding region and consequently increasing the relative salience of the retained region. These accompanying changes are not incidental side effects; they are the mechanisms through which each intervention reduces visual distractor burden. Clean causal identification of distractor count alone would instead require changing the number of distractors while holding target appearance, scale, position, background, and scene structure fixed.

To test whether the observed benefit is specific to grayscale suppression, we compare graying with two complementary operators, blur and inpainting, over the same retained regions. Blur suppresses the surrounding region without replacing it with grayscale values. Inpainting provides the closest practical approximation to selectively removing same-category distractors while leaving the remainder of the image unchanged.

For inpainting, the edit mask is defined as the intersection of the suppressed region with SAM3-segmented polygons of target-category instances. The original image, region guide, and mask are supplied to the Codex image-generation model, and the generated content is composited through the same binary mask. Pixels outside the edit mask are therefore preserved exactly. However, the masks depend on SAM3 and the generated content inside them may contain segmentation or inpainting artifacts, so this remains a practical approximation rather than an exact counterfactual intervention.

Qwen2.5-VL-7B, which was not part of the graying comparison, changes by +0.7 under blur and +0.3 under inpainting in the 1024 condition. Table 15 reports the 512×512 condition and Table 16 the 1024×1024 condition.

Table 15: Distractor-reduction interventions at 512×512 on the 800-sample subset. Base is unmodified-image $\mathcal{I}_{\mathrm{ACC}}$; intervention entries are changes in percentage points. Qwen2.5-VL-7B was not part of the graying comparison and is omitted.

Model	**Base**	**Gray 512**	**Blur 512**	**Inpaint 512**
Qwen3-VL-8B	39.2	+16.9	+10.1	+0.0
InternVL3-14B	36.1	+17.1	+12.7	+18.4
InternVL3.5-14B	35.4	+7.1	+10.1	+12.1
InternVL3-8B	37.3	+6.7	+6.3	+3.5
InternVL3.5-8B	39.2	+7.9	+4.2	+5.5
Mean	37.4	**+11.1**	**+8.7**	**+7.9**

At 512×512, the mean gains are 11.1 points for graying, 8.7 for blur, and 7.9 for inpainting; at 1024×1024, the corresponding gains are 5.8, 5.8, and 7.3 points. The model-level ranking varies, and Qwen3-VL-8B shows no gain under inpainting. Nevertheless, improvements under both blur and inpainting show that the observed benefit is not specific to grayscale highlighting. Taken together, these interventions support the broader interpretation that reducing visual distractor burden can improve MIMO performance, while not establishing a clean causal effect of distractor count alone.

Table 16: Distractor-reduction interventions at 1024×1024 on the same subset. Entries are changes in percentage points from the same unmodified-image baseline.

Model	Base	Gray 1024	Blur 1024	Inpaint 1024
Qwen3-VL-8B	39.2	+11.4	+6.6	+0.0
InternVL3-14B	36.1	+12.2	+9.9	+16.6
InternVL3.5-14B	35.4	+1.6	+6.5	+12.8
InternVL3-8B	37.3	+1.5	+1.9	+2.4
InternVL3.5-8B	39.2	+2.3	+4.1	+4.8
Mean	37.4	**+5.8**	**+5.8**	**+7.3**

C.7 Naive Relevant-Region Extraction: Mechanism and Controls

The zooming strategy discussed in Sec. 6 of the main paper operates via a two-turn MLLM inference pipeline. Here we describe the mechanism in detail.

Turn 1 — Region Proposal. The model is presented with the original image together with the original question. Rather than being asked to answer the question directly, the model is instructed to identify the *region of interest* needed to answer the question and output its spatial extent as a set of bounding-box coordinates (top-left and bottom-right corners, normalized to the image dimensions). The intuition is that the model, even without explicit crop-and-re-query, has sufficient global context to localize the relevant region within a dense, multi-instance scene.

Cropping. The bounding-box coordinates returned by the model in Turn 1 are used to crop the original image. This step is intended to (i) *reduce the search space* by discarding irrelevant portions of the image and (ii) *exclude same-category distractors* that fall outside the selected region. Cropping also changes target scale, effective resolution and the amount of surrounding context, so these factors are not varied independently.

Turn 2 — Answer Generation. The cropped image is fed to the model along with the original question in a second inference call. Because the crop retains only the selected region, the model now operates on a narrower visual context containing fewer candidate instances.

Interpretation. MIMO-Bench places multiple same-category instances in close proximity, making it easy for the model to attend to the wrong object. By committing to a single spatial region before answering, the naive strategy narrows the set of candidate instances the model must consider. As reported in Table 5 of the main paper, this two-turn strategy yields consistent accuracy gains across all tested models — for example, +16.90 points for IVL3-14B and +11.75 points for IVL3.5-14B . These gains indicate that successful target-region selection can substantially improve downstream performance in dense MIMO scenes; the experiment does not isolate distractor interference, localization, recognition and attribute binding as independent causal contributors.

We evaluate whether the relevant-region results can be explained by generic cropping, rescaling, or an additional inference call. Fixed and reproducibly sampled random windows are evaluated at 512×512 and 1024×1024. A target is visible when more than half of its ground-truth box lies inside the window. Table 17 reports these fixed and random crop controls.

Table 17: Fixed and random crop controls on the 800-sample intervention subset. Δ is the change from the matched full-image baseline; visibility columns condition accuracy on whether the target remains in the crop.

Control	Δ vs. Base	Target visible	Target absent	Visibility Δ
Fixed 1024	+1.0	49.7	34.1	**+15.6**
Random 1024	+0.8	53.5	31.9	**+21.6**
Fixed 512	-5.3	58.3	33.2	**+25.1**
Random 512	-8.7	49.0	31.3	**+17.7**

The 1024 crops are essentially unchanged overall, while the 512 crops reduce accuracy because they exclude the target more often. Within every condition, retaining the target is associated with a 15.6–25.1 point advantage. This direction holds for all 24 model-by-condition comparisons; when the target is absent, accuracy falls to approximately 31–34%, close to the majority-answer baseline. Because resolution is identical within each visibility split, resolution alone cannot explain this difference.

MIMO-Crop is an annotation-derived crop centered on the target that retains the objects needed to resolve the referring expression and is independent of the evaluated MLLM. On the full benchmark it raises mean accuracy from 39.5% to 48.2%. Table 18 reports the MIMO-Crop reference on the full benchmark.

Table 18: Full-benchmark performance with the externally supplied MIMO-Crop$_{512}$ relevant region. This is not a matched baseline for the 800-sample predicted-crop analysis.

Model	**Base**	**MIMO-Crop$_{512}$**
InternVL3-14B	37.2	50.7
InternVL3-8B	35.9	46.8
InternVL3.5-8B	40.5	50.7
Qwen3-VL-32B	47.5	48.9
Qwen3-VL-8B	40.3	44.4
Qwen2.5-VL-7B	35.8	47.6
Mean	**39.5**	**48.2**

We also condition performance on the IoU of model-selected regions with the ground-truth target box. Across predicted crops, the mean region is 459×479 pixels (12.43% of the original image area). The base and IoU columns use the 800-sample intervention subset. MIMO-Crop$_{512}$ uses the full benchmark and is shown as an external relevant-region reference rather than a matched subset baseline. The final column reports the change under naive model-driven zooming; for Qwen3-VL-32B, the corresponding subset accuracy is 46.3 before and 49.3 after zooming. Table 19 reports accuracy conditioned on predicted-crop IoU, and Table 20 the corresponding reference and zooming gain per model.

Table 19: Final-answer accuracy conditioned on predicted-crop IoU with the ground-truth target box. Base and IoU bins use the 800-sample intervention subset. All values are percentages.

Model	**Base**	**0–25**	**25–50**	**50–75**	**75–100**
InternVL3-14B	36.1	48.8	51.6	56.5	52.0
InternVL3-8B	37.3	35.4	44.9	51.5	51.0
InternVL3.5-8B	39.2	37.2	42.7	36.4	45.8
Qwen3-VL-32B	46.3	41.2	45.5	49.0	52.9
Qwen3-VL-8B	39.2	39.3	45.5	39.0	50.8
Qwen2.5-VL-7B	35.8	29.7	40.5	28.6	39.2
Mean	**39.0**	**38.6**	**45.1**	**43.5**	**48.6**

Table 20: Relevant-region reference and zooming gain for the same models. MIMO-Crop$_{512}$ is evaluated on the full benchmark and is therefore an external reference rather than a matched subset baseline. Zoom Δ is the change under naive model-driven relevant-region extraction.

Model	**MIMO-Crop$_{512}$**	**Zoom Δ**
InternVL3-14B	50.7	+16.90
InternVL3-8B	46.8	+5.46
InternVL3.5-8B	50.7	+2.14
Qwen3-VL-32B	48.9	+3.0
Qwen3-VL-8B	44.4	+0.89
Qwen2.5-VL-7B	47.6	+1.50
Mean	**48.2**	**+4.98**

Low-IoU regions (0–25%) achieve 38.6%, close to the 39.0% full-image baseline, whereas high-IoU regions (75–100%) reach 48.6%, close to the 48.2% MIMO-Crop$_{512}$ result. IoU is only an approximate measure of crop usefulness because a compositional RE can require neighboring context beyond the target box; we therefore interpret this as an association between better region selection and higher downstream accuracy.

Finally, a two-pass control keeps the region-proposal turn and second inference call but reuses the unmodified image in the second turn. Table 21 reports this two-pass control.

Table 21: Two-pass no-crop control on the 800-sample intervention subset. Δ is the change in $\mathcal{I}_{\mathrm{ACC}}$ from the matched single-pass full-image condition.

Model	Δ
Qwen2.5-VL-7B	+1.0
Qwen3-VL-8B	+1.5
InternVL3-8B	+2.0
InternVL3.5-8B	+1.1
InternVL3-14B	-1.5
InternVL3.5-14B	+3.5
Mean	**+1.3**

The second inference call alone yields only a 1.3 point mean gain, with no model gaining more than 3.5 points. Together, these controls show that generic cropping/rescaling by itself and a second call by itself do not account for the observed relevant-region gains; the benefit is associated with successfully retaining useful target-relevant information.

These experiments do not isolate localization, rescaling, distractor reduction, and additional inference as independent causal contributors. Here, "dynamic visual access" denotes a small exploratory single-pass narrowing study in which the first inference selects the visual region supplied in the second; it is not a complete iterative visual-search system. We present it as a proof of concept that re-accessing a useful region can improve MIMO reasoning and as motivation for future learned multi-step visual access.

C.8 Zero-Shot and Self-Consistency CoT: Inference Mechanism

Sec. 6 of the main paper ("Does CoT Enhance MIMO Performance?") evaluates Zero-Shot CoT (ZS-CoT) and Self-Consistency CoT (SC-CoT). Both build on the base prompts of Figs. 23 and 24.

ZS-CoT. The base prompt is appended with *"Let's think step by step,"* which precedes the model's answer. Decoding is greedy (`do_sample=False`), and a single generation is used to score accuracy.

SC-CoT. The same ZS-CoT prompt is reused, but generation is stochastic: `do_sample=True` with temperature 0.7, top-$k = 40$, top-$p = 1.0$. Five generations are sampled per question, and the final prediction is the **majority-vote** answer across them (ties broken by first-encountered order).

C.9 Uncertainty and Statistical Analysis

We report 95% nonparametric percentile bootstrap intervals using 20,000 replicates. For the base bootstrap intervals, we resample the unit on which each metric is defined: image–object–RE–attribute quadruplets for $\mathcal{I}_{\mathrm{ACC}}$, target groups for G_{ACC}, counting queries for $\mathcal{I}_{\mathrm{MRA}}$, and parent groups for $\mathcal{L}_{\mathrm{MRA}}$ and $\mathcal{G}_{\mathrm{MRA}}$. Because questions sharing an image, target object, or parent query are not independent, we evaluate progressively coarser resampling units. The central point estimate is unchanged across columns; only the bootstrap unit differs, with coarser units giving wider intervals.

Pooled scores use the metric-appropriate aggregation unit: the AB questions for $\mathcal{I}_{\mathrm{ACC}}$, target groups for G_{ACC}, the NC questions for $\mathcal{I}_{\mathrm{MRA}}$, and the parent groups for $\mathcal{L}_{\mathrm{MRA}}$ and $\mathcal{G}_{\mathrm{MRA}}$. Table 22 reports $\mathcal{I}_{\mathrm{ACC}}$ under sample-, object- and image-level resampling, and Table 23 the corresponding G_{ACC} intervals under target-group and image-level resampling. For numerical counting, Tables 24, 25 and 26 report $\mathcal{I}_{\mathrm{MRA}}$, $\mathcal{L}_{\mathrm{MRA}}$ and $\mathcal{G}_{\mathrm{MRA}}$ intervals at the units each metric is defined over.

Table 22: Attribute-binding $\mathcal{I}_{\mathrm{ACC}}$ point estimates and 95% bootstrap intervals under sample-, object-, and image-level resampling.

Model	Sample	Object	Image
Gemini-3.5 Flash Low	54.7 [52.0, 57.3]	54.7 [51.6, 57.9]	54.7 [49.7, 62.3]
Gemini-3.1 Pro Preview Low	55.0 [52.3, 57.7]	55.0 [51.8, 58.3]	55.0 [50.0, 62.4]
Gemini-2.5 Flash	50.2 [48.1, 52.4]	50.2 [47.8, 52.7]	50.2 [47.3, 53.7]
Gemini-2.5 Pro	48.5 [46.7, 50.3]	48.5 [46.3, 50.6]	48.5 [45.5, 51.6]
GPT-5 minimal	30.7 [26.8, 35.2]	30.7 [28.9, 33.1]	30.7 [29.1, 32.2]
GPT-5 medium	51.8 [49.9, 53.6]	51.8 [49.6, 53.9]	51.8 [48.6, 55.0]
GLM-4.5V	47.4 [44.7, 50.0]	47.4 [44.0, 50.9]	47.4 [38.7, 55.4]
InternVL3.5-241B-A28B	46.2 [44.3, 48.0]	46.2 [43.9, 48.3]	46.2 [43.1, 49.5]
InternVL3-78B	39.3 [37.7, 41.0]	39.3 [37.4, 41.2]	39.3 [36.3, 42.8]
Qwen3-VL-32B-Instruct	47.5 [45.8, 49.2]	47.5 [45.6, 49.4]	47.5 [44.9, 50.2]
Qwen3-VL-8B-Instruct	40.3 [38.6, 42.1]	40.3 [38.5, 42.2]	40.3 [38.0, 42.8]
Qwen3-VL-30B-A3B-Instruct	33.8 [31.9, 35.6]	33.8 [31.7, 35.9]	33.8 [28.9, 38.3]
Qwen2.5-VL-72B-Instruct	42.6 [39.5, 45.7]	42.6 [38.7, 46.6]	42.6 [36.3, 49.9]
Qwen2.5-VL-32B-Instruct	39.3 [36.7, 41.9]	39.3 [36.3, 42.5]	39.3 [35.7, 43.8]
Qwen2.5-VL-7B-Instruct	35.8 [33.8, 37.8]	35.8 [33.7, 37.9]	35.8 [32.5, 39.1]
InternVL3.5-38B	42.8 [40.8, 44.8]	42.8 [40.5, 45.0]	42.8 [40.0, 45.9]
InternVL3.5-14B	34.4 [32.8, 36.0]	34.4 [32.7, 36.1]	34.4 [31.4, 37.6]
InternVL3.5-8B	40.5 [38.9, 42.1]	40.5 [38.8, 42.1]	40.5 [37.1, 44.4]
InternVL3-38B	44.3 [42.5, 46.1]	44.3 [42.2, 46.5]	44.3 [39.0, 51.3]
InternVL3-14B	37.2 [35.5, 38.9]	37.2 [35.3, 39.0]	37.2 [34.7, 40.0]
InternVL3-8B	35.9 [34.3, 37.5]	35.9 [34.1, 37.7]	35.9 [33.1, 38.5]
GLM-4.1V-9B-Thinking	36.0 [33.4, 38.5]	36.0 [33.5, 38.5]	36.0 [32.5, 38.9]

Table 23: Attribute-binding G_{ACC} point estimates and 95% bootstrap intervals under target-group and image-level resampling.

Model	Group	Image
Gemini-3.5 Flash Low	62.3 [59.0, 65.6]	62.3 [56.0, 69.7]
Gemini-3.1 Pro Preview Low	61.8 [58.4, 65.3]	61.8 [54.1, 69.3]
Gemini-2.5 Flash	46.3 [43.7, 48.9]	46.3 [43.9, 51.4]
Gemini-2.5 Pro	46.9 [44.2, 49.5]	46.9 [43.7, 55.3]
GPT-5 minimal	27.2 [24.3, 31.2]	27.2 [23.1, 30.2]
GPT-5 medium	50.4 [47.9, 52.9]	50.4 [46.4, 54.4]
GLM-4.5V	40.8 [36.9, 44.6]	40.8 [37.0, 47.2]
InternVL3.5-241B-A28B	43.9 [41.3, 46.5]	43.9 [39.8, 47.7]
InternVL3-78B	37.3 [35.0, 39.6]	37.3 [32.3, 44.2]
Qwen3-VL-32B-Instruct	46.5 [44.5, 48.6]	46.5 [43.3, 50.6]
Qwen3-VL-8B-Instruct	36.6 [34.6, 38.7]	36.6 [34.8, 41.9]
Qwen3-VL-30B-A3B-Instruct	29.7 [27.3, 32.0]	29.7 [26.2, 35.4]
Qwen2.5-VL-72B-Instruct	42.3 [38.1, 46.5]	42.3 [31.1, 50.2]
Qwen2.5-VL-32B-Instruct	39.2 [35.9, 42.4]	39.2 [35.7, 45.2]
Qwen2.5-VL-7B-Instruct	34.7 [32.2, 37.2]	34.7 [31.3, 37.7]
InternVL3.5-38B	42.7 [40.2, 45.3]	42.7 [39.6, 48.0]
InternVL3.5-14B	36.5 [34.7, 38.4]	36.5 [34.2, 41.2]
InternVL3.5-8B	41.3 [39.4, 43.2]	41.3 [37.3, 42.9]
InternVL3-38B	43.8 [41.4, 46.2]	43.8 [37.4, 46.0]
InternVL3-14B	37.3 [35.2, 39.4]	37.3 [33.9, 41.9]
InternVL3-8B	36.5 [34.6, 38.4]	36.5 [33.5, 39.3]
GLM-4.1V-9B-Thinking	35.7 [32.6, 38.8]	35.7 [32.0, 40.0]

Table 24: Numerical-counting $\mathcal{I}_{\mathrm{MRA}}$ point estimates and 95% bootstrap intervals under sample-, parent-group-, and image-level resampling.

Model	Sample	Group	Image
Gemini-3.5 Flash Low	61.5 [56.3, 66.5]	61.5 [54.3, 68.3]	61.5 [49.7, 72.1]
Gemini-3.1 Pro Preview Low	61.3 [56.0, 66.5]	61.3 [54.4, 67.9]	61.3 [50.3, 71.6]
Gemini-2.5 Flash	40.8 [35.0, 46.5]	40.8 [34.2, 47.5]	40.8 [30.9, 51.2]
Gemini-2.5 Pro	49.0 [43.5, 54.5]	49.0 [42.0, 56.0]	49.0 [39.9, 57.9]
GPT-5 minimal	44.0 [38.6, 49.3]	44.0 [37.4, 50.6]	44.0 [34.5, 53.5]
GPT-5 medium	49.0 [43.8, 54.2]	49.0 [42.8, 55.3]	49.0 [41.1, 56.6]
GLM-4.5V	42.3 [37.0, 47.7]	42.3 [36.4, 48.2]	42.3 [33.1, 51.3]
InternVL3.5-241B-A28B	49.9 [45.9, 54.4]	49.9 [42.3, 59.0]	49.9 [37.4, 65.1]
InternVL3-78B	37.5 [32.2, 42.9]	37.5 [31.0, 44.3]	37.5 [28.0, 47.6]
Qwen3-VL-32B-Instruct	47.7 [42.3, 53.1]	47.7 [41.1, 54.3]	47.7 [38.2, 57.2]
Qwen3-VL-8B-Instruct	42.6 [37.0, 48.2]	42.6 [35.9, 49.2]	42.6 [33.2, 52.1]
Qwen3-VL-30B-A3B-Instruct	35.0 [29.8, 40.3]	35.0 [28.4, 41.9]	35.0 [26.3, 43.4]
Qwen2.5-VL-72B-Instruct	40.4 [35.0, 45.7]	40.4 [34.1, 46.8]	40.4 [30.3, 50.7]
Qwen2.5-VL-32B-Instruct	37.3 [32.1, 42.5]	37.3 [30.9, 43.8]	37.3 [27.6, 47.8]
Qwen2.5-VL-7B-Instruct	31.4 [26.6, 36.4]	31.4 [25.0, 38.0]	31.4 [21.2, 42.4]
InternVL3.5-38B	36.8 [31.6, 42.1]	36.8 [30.6, 43.1]	36.8 [28.8, 44.5]
InternVL3.5-14B	38.2 [32.9, 43.6]	38.2 [32.3, 44.1]	38.2 [28.4, 48.5]
InternVL3.5-8B	41.2 [35.7, 46.7]	41.2 [35.5, 47.2]	41.2 [32.3, 51.0]
InternVL3-38B	41.0 [35.5, 46.6]	41.0 [34.4, 47.6]	41.0 [32.9, 48.7]
InternVL3-14B	33.2 [26.8, 39.8]	33.2 [26.1, 41.0]	33.2 [24.0, 42.4]
InternVL3-8B	33.3 [27.5, 39.0]	33.3 [27.3, 38.4]	33.3 [26.5, 39.4]
GLM-4.1V-9B-Thinking	25.8 [16.8, 35.3]	25.8 [17.6, 34.7]	25.8 [17.4, 34.3]

Table 25: Numerical-counting $\mathcal{L}_{\mathrm{MRA}}$ point estimates and 95% bootstrap intervals under parent-group and image-level resampling.

Model	Group	Image
Gemini-3.5 Flash Low	60.8 [52.2, 68.9]	60.8 [47.6, 73.0]
Gemini-3.1 Pro Preview Low	60.5 [52.2, 68.3]	60.5 [51.0, 70.1]
Gemini-2.5 Flash	37.1 [28.9, 45.5]	37.1 [27.0, 48.0]
Gemini-2.5 Pro	44.0 [35.7, 52.6]	44.0 [32.2, 55.8]
GPT-5 minimal	42.9 [35.1, 50.7]	42.9 [32.8, 52.9]
GPT-5 medium	46.7 [40.0, 53.7]	46.7 [37.7, 55.9]
GLM-4.5V	44.6 [37.8, 51.5]	44.6 [35.0, 54.4]
InternVL3.5-241B-A28B	50.4 [42.1, 60.1]	50.4 [33.9, 69.6]
InternVL3-78B	37.1 [29.5, 44.9]	37.1 [28.1, 45.9]
Qwen3-VL-32B-Instruct	48.7 [40.8, 56.5]	48.7 [38.0, 59.1]
Qwen3-VL-8B-Instruct	42.4 [34.6, 50.3]	42.4 [30.8, 53.9]
Qwen3-VL-30B-A3B-Instruct	41.6 [34.0, 49.6]	41.6 [31.2, 52.9]
Qwen2.5-VL-72B-Instruct	43.5 [36.9, 50.2]	43.5 [34.5, 52.2]
Qwen2.5-VL-32B-Instruct	36.3 [29.0, 43.6]	36.3 [26.8, 46.4]
Qwen2.5-VL-7B-Instruct	32.4 [24.8, 40.2]	32.4 [22.6, 42.7]
InternVL3.5-38B	41.5 [33.5, 49.4]	41.5 [30.9, 51.7]
InternVL3.5-14B	38.5 [31.2, 46.1]	38.5 [29.9, 46.6]
InternVL3.5-8B	42.6 [35.1, 50.2]	42.6 [34.9, 50.0]
InternVL3-38B	46.5 [38.3, 55.0]	46.5 [36.5, 57.0]
InternVL3-14B	34.1 [25.6, 43.3]	34.1 [24.4, 43.8]
InternVL3-8B	39.8 [22.6, 46.2]	39.8 [28.4, 47.7]
GLM-4.1V-9B-Thinking	30.2 [19.4, 41.9]	30.2 [18.8, 41.7]

Table 26: Numerical-counting $\mathcal{G}_{\mathrm{MRA}}$ point estimates and 95% bootstrap intervals under sample- and image-level resampling.

Model	Sample	Image
Gemini-3.5 Flash Low	60.7 [52.5, 68.5]	60.7 [50.1, 70.0]
Gemini-3.1 Pro Preview Low	63.8 [55.6, 71.7]	63.8 [53.2, 73.0]
Gemini-2.5 Flash	47.3 [37.1, 57.5]	47.3 [32.7, 61.4]
Gemini-2.5 Pro	66.4 [58.0, 74.5]	66.4 [56.8, 76.1]
GPT-5 minimal	43.8 [34.8, 52.8]	43.8 [31.8, 55.8]
GPT-5 medium	55.1 [46.1, 63.8]	55.1 [42.2, 67.2]
GLM-4.5V	38.5 [30.0, 46.9]	38.5 [27.1, 49.5]
InternVL3.5-241B-A28B	46.8 [44.3, 51.7]	46.8 [44.7, 51.0]
InternVL3-78B	35.6 [26.5, 45.1]	35.6 [22.0, 49.8]
Qwen3-VL-32B-Instruct	51.4 [42.5, 60.4]	51.4 [39.4, 63.5]
Qwen3-VL-8B-Instruct	38.0 [28.4, 48.4]	38.0 [25.3, 51.3]
Qwen3-VL-30B-A3B-Instruct	32.9 [23.6, 42.4]	32.9 [21.9, 44.5]
Qwen2.5-VL-72B-Instruct	36.8 [26.8, 46.8]	36.8 [23.5, 50.2]
Qwen2.5-VL-32B-Instruct	41.2 [32.4, 50.0]	41.2 [28.6, 53.8]
Qwen2.5-VL-7B-Instruct	28.6 [20.5, 37.0]	28.6 [17.6, 41.1]
InternVL3.5-38B	35.1 [26.1, 44.2]	35.1 [25.1, 44.8]
InternVL3.5-14B	37.8 [28.6, 46.9]	37.8 [25.2, 50.4]
InternVL3.5-8B	35.1 [25.4, 44.7]	35.1 [23.3, 47.0]
InternVL3-38B	33.4 [24.5, 42.7]	33.4 [19.5, 47.7]
InternVL3-14B	30.0 [18.6, 41.7]	30.0 [16.5, 44.6]
InternVL3-8B	18.4 [10.4, 27.0]	18.4 [5.6, 25.3]
GLM-4.1V-9B-Thinking	29.5 [13.3, 48.7]	29.5 [9.5, 53.9]

Coarser clustering generally widens the intervals by approximately 1.3–1.8 times, consistent with within-image dependence, although the ordering is not monotone for every model.

For matched intervention conditions, we preserve example-level pairing and report the paired change with its bootstrap interval. Visual prompting replaces the textual RE with a ground-truth visual cue. We define the paired visual-prompting change as

$$\Delta_{\mathrm{VP}} = \mathcal{I}_{\mathrm{ACC}}(\text{with visual prompting}) - \mathcal{I}_{\mathrm{ACC}}(\text{without visual prompting}).$$

Table 27 reports these paired visual-prompting changes.

Table 27: Paired change in $\mathcal{I}_{\mathrm{ACC}}$ under visual prompting on the matched 800-sample subset. All intervals exclude zero.

Model	$\Delta\ \mathcal{I}_{\mathrm{ACC}}$	95% CI of Δ
Qwen3-VL-8B-Instruct	**+25.5**	[19.84, 31.59]
GLM-4.5V	**+18.5**	[13.42, 23.96]
Qwen3-VL-32B-Instruct	**+18.0**	[13.85, 22.12]
Qwen2.5-VL-32B-Instruct	**+22.5**	[18.11, 26.98]
Qwen3-VL-30B-A3B-Instruct	**+24.0**	[16.20, 29.15]
InternVL3-38B	**+10.8**	[6.95, 14.97]
InternVL3.5-38B	**+12.4**	[8.04, 17.17]
InternVL3-14B	**+13.2**	[9.81, 17.01]
InternVL3.5-14B	**+11.2**	[7.09, 15.44]
InternVL3-8B	**+8.6**	[5.29, 12.37]
InternVL3.5-8B	**+6.0**	[3.87, 8.48]

Every visual-prompting interval excludes zero. The positive result also survives image-clustered two-sided sign-flip tests with Benjamini–Hochberg correction ($q \leq 0.0312$ for $\mathcal{I}_{\mathrm{ACC}}$ and $q \leq 0.0202$ for G_{ACC}).

For paired significance, condition labels are exchanged jointly within the corresponding image or object group, and Benjamini–Hochberg FDR correction is applied within each family of model comparisons. We attach randomization-test p-values to visual prompting, for which we make a positive claim. For reasoning and Chain-of-Thought, the paired intervals overwhelmingly include zero and already support the stated non-improvement conclusion, so additional randomization-test values would only restate that result. Ta-

ble 28 reports the paired reasoning-mode changes for attribute binding and Table 29 for numerical counting; Table 30 reports the corresponding chain-of-thought changes.

Table 28: Paired reasoning-mode changes and 95% bootstrap intervals for attribute binding. For Qwen models, Δ is Thinking minus Instruct; for Gemini models, it is High minus Low reasoning effort. Bold entries have intervals excluding zero.

Model	$\Delta\mathcal{I}_{\mathrm{ACC}}$ (95% CI)	ΔG_{ACC} (95% CI)
Q3-8B	**+5.2 [+0.8, +9.9]**	+4.7 [-2.7, +12.1]
Q3-30B-A3B	+1.8 [-0.9, +4.4]	**+5.0 [+1.9, +7.6]**
Q3-32B	-0.1 [-4.4, +7.4]	-2.5 [-9.5, +10.5]
G3.1-Pro-Preview	-0.6 [-2.5, +1.1]	+0.1 [-2.4, +2.4]
G3.5-Flash	**+3.6 [+0.6, +5.7]**	-1.9 [-4.8, +0.4]

Table 29: Paired reasoning-mode changes and 95% bootstrap intervals for numerical counting, using the same reasoning-mode comparison. Bold entries have intervals excluding zero.

Model	$\Delta\mathcal{I}_{\mathrm{MRA}}$ (95% CI)	$\Delta\mathcal{L}_{\mathrm{MRA}}$ (95% CI)	$\Delta\mathcal{G}_{\mathrm{MRA}}$ (95% CI)
Q3-8B	+3.4 [-6.0, +13.5]	**+22.3 [+11.2, +36.2]**	+5.9 [-4.4, +15.3]
Q3-30B-A3B	+1.4 [-4.5, +3.1]	+0.4 [-3.9, +3.3]	+0.8 [-6.2, +7.6]
Q3-32B	+2.5 [-8.9, +11.9]	+2.8 [-10.4, +14.3]	-6.2 [-23.1, +7.6]
G3.1-Pro-Preview	-4.8 [-8.6, +1.3]	-5.9 [-12.4, +4.1]	+1.2 [-10.7, +11.9]
G3.5-Flash	**+6.4 [+3.2, +10.6]**	**+13.3 [+9.3, +18.0]**	**+5.0 [+0.4, +12.3]**

Table 30: Paired changes under zero-shot and self-consistency chain-of-thought prompting relative to the same base condition. Bold entries have 95% intervals excluding zero.

Model	Δ ZS-CoT vs. Base (95% CI)	Δ SC-CoT vs. Base (95% CI)
IVL3-14B	**+4.0 [+0.8, +7.5]**	+6.0 [-2.3, +11.6]
IVL3-8B	-1.0 [-4.0, +2.3]	-1.0 [-5.1, +6.5]
IVL3.5-14B	+1.0 [-4.3, +5.4]	+0.0 [-14.2, +9.1]
IVL3.5-8B	**+7.0 [+4.3, +9.6]**	+3.0 [-4.8, +8.3]
QVL2.5-7B	+0.0 [-4.9, +6.0]	+1.0 [-14.8, +9.6]
QVL2.5-32B	-2.0 [-13.2, +6.7]	-3.0 [-10.1, +2.2]
QVL3-8B	+4.0 [-5.0, +8.9]	+4.0 [-8.4, +12.0]
QVL3-32B	-2.0 [-12.3, +3.6]	-6.0 [-10.3, +1.3]

Reasoning effort does not improve every model–metric combination, and most CoT intervals contain zero. The only CoT intervals excluding zero are positive zero-shot effects for IVL3-14B and IVL3.5-8B. For clustered global-versus-local counting, only Qwen3-VL-8B-Thinking remains significant after correction ($\Delta_{G-L} = -24.6$, 95% CI $[-39.9, -12.8]$, $p = 0.0020$, $q = 0.0488$).

Six participants who did not annotate MIMO-Bench each completed 25 AB and 15 NC groups. Human intervals resample groups rather than treating individual questions as independent. Table 31 reports the human point estimates and group-resampled intervals.

Table 31: Human baseline point estimates and group-resampled 95% bootstrap intervals. The lowest lower bound (77.1%) remains above the strongest corresponding model result.

Metric	**Human**	**95% CI**
Real $\mathcal{I}_{\mathrm{ACC}}$	89.3	[81.7,93.2]
Real G_{ACC}	86.7	[83.2,89.8]
Syn. $\mathcal{I}_{\mathrm{ACC}}$	85.7	[77.1,89.6]
Syn. G_{ACC}	87.5	[84.9,90.1]
Real $\mathcal{I}_{\mathrm{MRA}}$	87.7	[85.1,88.2]
Real $\mathcal{L}_{\mathrm{MRA}}$	86.4	[83.2,88.5]
Real $\mathcal{G}_{\mathrm{MRA}}$	90.1	[87.9,92.6]
Syn. $\mathcal{I}_{\mathrm{MRA}}$	89.6	[86.9,91.9]
Syn. $\mathcal{L}_{\mathrm{MRA}}$	87.4	[85.7,90.2]
Syn. $\mathcal{G}_{\mathrm{MRA}}$	92.0	[90.1,94.4]

D Limitations

We discuss some of the limitations of our work.

- Scaling MIMO-Bench is challenging because of its annotation pipeline. The entire process from bounding box and referring expression creation to question answering and verification relies heavily on humans. Our attempts to semi-automate the generation of referring expressions were unsuccessful, as existing models failed to produce unambiguous descriptions.
- Currently, we attribute the difficulty of MIMO-Bench to the joint effect of same-category distractors and the referring-expression requirement, without isolating either as the dominant factor. Distractors increase the difficulty of the underlying visual reasoning task, while the referring-expression requirement introduces multiple subtasks that the model must solve in order to correctly localize the target object. However, beyond the mere presence of distractors, their distribution within the scene also plays a significant role in influencing model performance. In future work, we aim to study how the number and spatial arrangement of distractors jointly affect the performance of MLLMs.