ZOOMINC: MINING GLOBAL-TO-ZOOM PERCEPTION GAPS FOR FINE-GRAINED VISUAL SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained visual perception requires Vision-Language Models (VLMs) to locate and interpret subtle evidence within complex scenes. Constructing supervision for this ability is challenging: questions must have correct, unambiguous answers that depend on actual fine-grained visual details rather than coarse cues or language priors. Verifying these properties makes manual annotation labor-intensive and automatic generation difficult for models that themselves struggle with fine-grained perception. Yet the same models can often recognize such details once the region is zoomed in. We exploit this global-to-zoom perception gap and introduce **ZoomInc**, a data construction pipeline that turns **inc**remental details revealed *only* by **zoom**ing into questions about the full image. It verifies each answer on the zoomed view with multiple models, ensures that each question remains unambiguous in the full image and cannot be answered from language priors, and specifically generates high-quality distractors for multiple-choice questions. With this pipeline and human-only screening (no manual QA editing), we build ZoomInc Bench, a 1,000-question bilingual benchmark on which the best of 28 VLMs achieves only 61.08. Failure analysis reveals that models favor a particular type of distractor when answering multiple-choice questions incorrectly. Beyond evaluation, training on just 9K ZoomInc samples substantially improves the visual perception capabilities of our 4B/9B models, enabling them to outperform several hundred-billion-parameter models across various perception benchmarks. The trained 9B model achieves SoTA performance on several benchmarks, such as 90.25 on HR-Bench 4K. These results establish global-to-zoom perception gaps as a practical basis for scaling fine-grained visual supervision for both evaluation and training. Pipeline code and data will be released after publication.

1 INTRODUCTION

Fine-grained visual perception is essential in many practical applications, from detecting small vehicles in aerial imagery Xia et al. (2018) and identifying subtle defects in industrial inspection Bergmann et al. (2019) to reading product labels for people with visual impairments Gurari et al. (2018). Addressing such tasks with Vision-Language Models (VLMs) Team (2026); Bai et al. (2025); Zhu et al. (2025); Liu et al. (2023) requires locating and interpreting small details that are easily overlooked in the full image. To develop and benchmark such fine-grained capabilities, high-quality supervision is essential: answers must be correct and unambiguous, and questions must genuinely require fine-grained visual evidence.

Constructing such supervision at scale is challenging. Manual annotation is labor-intensive, as annotators must inspect small regions and verify each of these properties. Automatic generation Wang et al. (2025a); Wei et al. (2026) faces a related difficulty: the models used to generate questions may themselves struggle to perceive the relevant details in the full image. As a result, they may produce incorrect answers, ambiguous references, or questions that do not actually require fine-grained perception. Motivated by this, the key challenge is to obtain reliable supervision on visual details that models fail to perceive in full scenes.

Zooming in has long been used to recover such details, either at inference time through visual search Wu & Xie (2023); Lai et al. (2025); Shen et al. (2025) and agentic VLMs with cropping tools Zheng et al. (2026); Hong et al. (2026), or during training through region-to-image distilla-

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

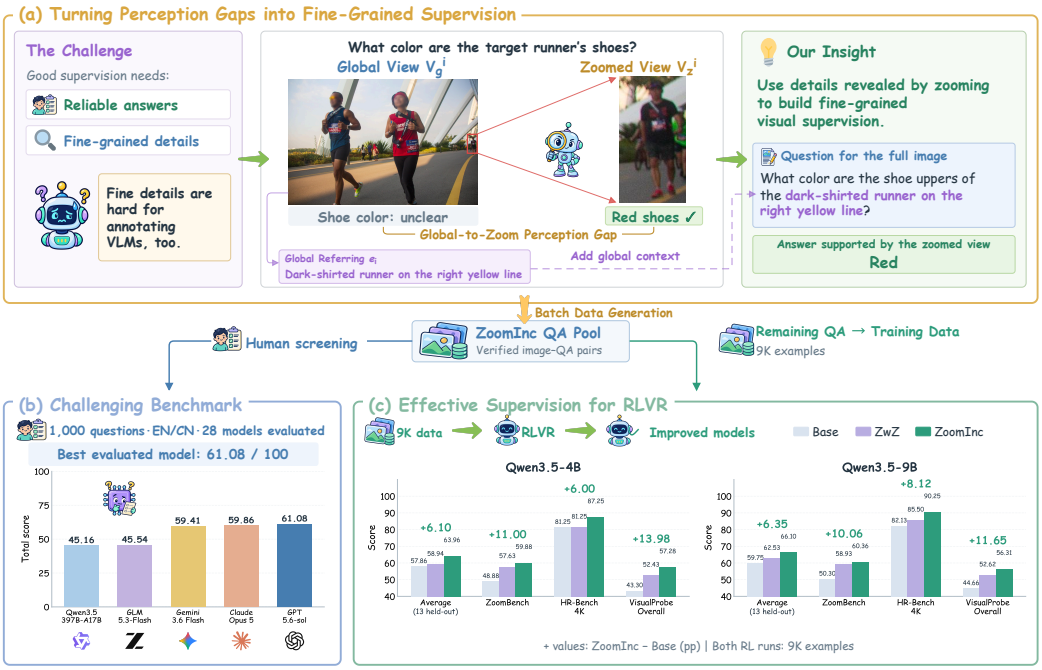

Figure 1: **ZoomInc: turning global-to-zoom perception gaps into fine-grained visual supervision.** (a) The runner's shoe color is unclear in the global view but discernible in the zoomed view; ZoomInc asks about such details in the full image, using a global referring expression to identify the target. The paired views V_g^i and V_z^i, and the referring e_i are defined in Section 3.2. (b) ZoomInc-Bench, obtained by human screening without QA editing, challenges all 28 evaluated VLMs (c) RLVR on 9K training QA pairs improves Qwen3.5-4B and Qwen3.5-9B over their base models and ZwZ-trained counterparts (ICML'26 Wei et al. (2026)).

tion Wei et al. (2026); Yuan et al. (2026), since details a model fails to resolve in the full image often become recognizable once the region is zoomed in. We specifically use this global-to-zoom perception gap to construct the supervision itself (Fig. 1 (a)): contrasting the two views locates details that are hard to perceive in the full image.

Based on this idea, we introduce **ZoomInc**, an automated data construction pipeline that converts **inc**remental details, which are revealed *only* by **zoom**ing, into challenging questions about the full image (**Contribution #1**). ZoomInc (1) builds questions exclusively from details described in the zoomed view but unclear or absent in the full-image description, so that answering requires the fine detail itself; (2) keeps an answer only when multiple models agree on it in the zoomed view, and discards questions with ambiguous full-image references or answerable without images; and (3) generates various multiple-choice distractors that could mislead the queried VLMs.

Using ZoomInc, we curate **ZoomInc Bench**, a bilingual benchmark of 1,000 questions on attribute recognition, counting, and OCR, whose questions and answers are entirely model-authored: two human reviewers only accept/reject candidates (**Contribution #2**). Among 28 evaluated VLMs, the best model, GPT-5.6-Sol, scores only 61.08, and only five exceed 50 (Fig. 1(b)). Error analysis further reveals that incorrect choices concentrate on distractors mined from full-image mispredictions during construction (Sec. 3.3). This indicates that when details are unresolved, models tend to fill the information gap with similar scene-plausible guesses, suggesting that the global-to-zoom perception gap is a common weakness across current VLMs, rather than an issue to specific models.

Beyond evaluation, ZoomInc data also serves as effective training supervision (**Contribution #3**). With identical source images and training settings, we train Qwen3.5-4B and Qwen3.5-9B Team (2026) on 9K ZoomInc and ZwZ (ICML'26 Wei et al. (2026)) QA pairs. ZoomInc-trained models consistently outperform ZwZ across all 14 evaluated benchmarks (Fig. 1(c)). The gains extend beyond fine-grained perception to general perception and difficult perception-and-reasoning tasks,

e.g., 11.01 points boost on O3-Bench Li et al. (2026a) for the 9B model. Notably, our fine-tuned 9B model is on par with GLM-5.3-Flash GLM-5-Team et al. (2026) and 4B model surpasses Qwen3.5-122B-A10B. Ablations show that removing either incremental evidence or multi-source distractors substantially reduces these gains, confirming that both designs rooted in the global-to-zoom perception gap are essential.

2 Related Work

Fine-grained perception benchmarks. A variety of benchmarks specifically evaluate fine-grained perception in high-resolution images, including Wu & Xie (2023); Wang et al. (2025b); Zhang et al. (2024); Wang et al. (2025a); Gavrikov et al. (2026); Lai et al. (2025); Zhang et al. (2026b); Wei et al. (2026); O3-Bench Li et al. (2026a) further requires multi-step reasoning beyond perception. However, questions in these benchmarks are typically annotated manually or generated by models with extensive manual editing (Table 1), making scalable cruation costly. Moreover, existing benchmarks rarely filter out language-prior shortcuts Cadene et al. (2019); Chen et al. (2024); Yue et al. (2025) or design targeted distractors. In contrast, ZoomInc fully automates QA authoring with zoom-view verification, blind shortcut filtering, and distractor mining from model errors.

Improving fine-grained perception. Zooming has been widely adopted to enhance fine-grained perception, either at inference time via visual search Wu & Xie (2023); Shen et al. (2025); Lai et al. (2025); Li et al. (2026a); Wang et al. and agentic cropping tools Zheng et al. (2026); Hong et al. (2026); Su et al. (2026); Zhang et al. (2025), or during training through zoomed-to-global distillation Wei et al. (2026); Yuan et al. (2026); Li et al. (2026b); Zhang et al. (2026a). Unlike prior works that focus on inference procedures or training objectives, ZoomInc targets data construction itself: it leverages the global-to-zoom perception gap to identify what to ask (strictly on zoom-revealed details) and reliably verify ground truth, making it complementary to existing approaches.

3 ZoomInc Data Construction Pipeline

ZoomInc exploits the perception gap between global and zoomed-in views: fine-grained details that are difficult to discern in the full image may become clear once the target region is zoomed in. As illustrated in Figure 2, ZoomInc first discovers and grounds target anchors (Sec. 3.1), then constructs questions and answers from details newly revealed by zooming (Sec. 3.2), and finally verifies answer reliability and visual dependence and builds multi-source distractors that capture plausible model errors (Sec.n 3.3). The complete algorithm and additional implementation details are provided in Appendix A.

3.1 Anchor Discovery and Grounding

Given an input image, denoted as the global view V_g, we first use a VLM to generate a global scene caption C_g. A text-only LLM then takes C_g as its sole input and proposes a set of candidate anchors. Each anchor i is represented by a referring expression e_i that describes the target's appearance, spatial location, or relations to other objects. To ensure that e_i can identify a localizable instance unambiguously in the full image, three VLMs independently ground e_i in V_g. We retain e_i only if each model returns exactly one valid bounding box, since a missing box or multiple boxes indicate that the referent cannot be found or may be ambiguous, and all pairwise box IoUs are at least 0.5, which provides cross-model evidence that e_i refers to a consistent region. For each retained anchor, we take the coordinate-wise median of the predicted boxes, expand the resulting region to include more surrounding context, yielding the crop box b_i, and crop V_g using b_i to obtain a zoomed view V_z^i. The quadruples (V_g, V_z^i, b_i, e_i) serve as inputs to the subsequent question construction stage.

3.2 Global-to-Zoom Evidence and QA Construction

For each retained anchor i, we use a strong VLM M_{cap} to describe the target region from both views. Specifically, we draw a red bounding box at b_i in V_g to obtain an annotated global view V_g^i. Figure 1(a) illustrates the paired views (V_g^i, V_z^i). We feed V_g^i into M_{cap} and prompt it to describe only the contents of the red box in as much detail as possible, yielding the global-view caption C_g^i.

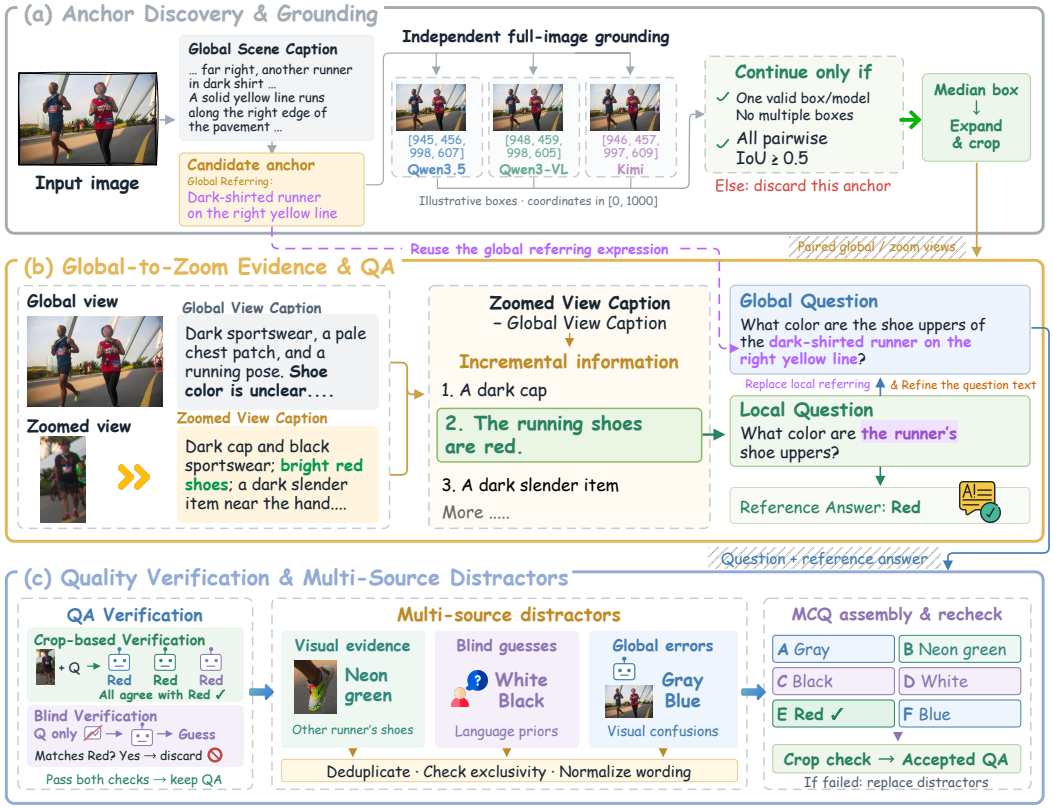

Figure 2: **Overview of ZoomInc.** (a) Discover anchors and obtain consensus regions through independent grounding. (b) Mine global-to-zoom incremental evidence from paired views to construct questions and reference answers. (c) Verify answers and visual dependence, then assemble and recheck multiple-choice questions with multi-source distractors.

In a separate, independent call, we provide the zoomed view V_z^i and request a detailed caption C_z^i. This produces two descriptions of the same region b_i at different viewing scales. We then use an LLM to compare C_g^i and C_z^i, extracting details that are explicitly described in C_z^i but remain unclear or unmentioned in C_g^i. These details form a candidate fact set Δ_i. For example, in Figure 2(b), the runner's shoe color is unclear in the global-view caption but identified as red in the zoomed-view caption, so it becomes a candidate fact in Δ_i.

We generate three types of questions: attribute recognition, text recognition (OCR), and counting. An LLM selects one fact from Δ_i that supports a definite answer and generates a local question q_{local}^i and a reference answer a_i based on this fact. The local question is phrased for the zoomed view with a simple reference such as "the runner", without specifying the target's location in the full image. The LLM then replaces this reference with the global referring expression e_i and adjusts the wording to produce a global-view question q_{global}^i, keeping the queried detail and reference answer unchanged, so that questions about details revealed by zooming clearly identify the intended target in the full image. For example, "What color are the runner's shoe uppers?" becomes "What color are the shoe uppers of the dark-shirted runner on the right yellow line?" Appendix H illustrates this construction process with attribute, counting, and OCR examples.

3.3 Quality Verification and Multi-Source Distractors

We verify each candidate QA pair in two ways. (1) For *crop-based verification*, we provide several VLMs with q_{local}^i and the zoomed view V_z^i, and sample multiple answers from each model. We retain the QA pair only if all models meet a predefined accuracy threshold on their sampled answers. Checking agreement among multiple models answering based on the zoomed-view helps ensure answer correctness, thereby avoiding sole reliance on a single model's caption for answer annotation,

which may contain hallucinations Rohrbach et al. (2018); Li et al. (2023); Jing et al. (2024). (2) For *blind verification*, models are asked to answer q^i_{global} without an image. We discard a QA if all blind models exceed a prescribed correct-answer threshold. This step effectively filters out questions that can be answered through language priors alone rather than actual image content Cadene et al. (2019); Chen et al. (2024); Yue et al. (2025).

For verified QA pairs suited to multiple-choice evaluation, we construct distractors from three complementary sources. (1) *Visual evidence* supplies plausible alternatives from image content, such as the neon-green shoes of another runner in Figure 2(c), which can expose errors in target selection or attribute binding. (2) *Blind guesses* provide incorrect answers produced without the image, capturing language priors. (3) *Global-view errors* provide incorrect answers produced from the full image, capturing confusions when local details are difficult to discern. We remove candidates equivalent to the reference answer or also valid for the question, merge semantically redundant candidates, and normalize their wording to form distinct answer options.

We combine the reference answer with the selected distractors and recheck the resulting MCQ using the zoomed view. This final check tests whether the reference answer remains reliably distinguishable after competing options are introduced. If the per-model agreement thresholds are not met, we replace or remove problematic distractors. OCR and counting questions retain an open-ended format and bypass distractor construction and MCQ rechecking. Appendix I provides examples of candidate QA pairs rejected by crop-based and blind verification.

4 Implementation Details and Data Preparation

Model Roles and Verification Settings. We use Qwen3.5-397B-A17B Team (2026) as M_{cap} and the initial global scene captioner. GLM-5.2 GLM-5-Team et al. (2026) performs incremental fact extraction and QA generation. Global-view error distractors are mined from incorrect answers produced by Qwen3.5-397B-A17B, Qwen3-VL-235B-A22B-Instruct Bai et al. (2025), and Kimi-K2.6 Team et al. (2026b) to the global question on the full image, with 10 answers sampled per model (30 in total). All verification stages likewise sample 10 answers per model and apply per-model accuracy thresholds. Complete model assignments, verification thresholds, and distractor repair settings are provided in Appendix B.

Data Generation and Benchmark Curation. We randomly sample 12K images from ZwZ Wei et al. (2026), which collects high-resolution images from several sources Kirillov et al. (2023); Chuang et al. (2025); Chen et al. (2022); Changpinyo et al. (2021); Krishna et al. (2017); Schuhmann et al. (2022), and generate a pool of 23,610 QA pairs with ZoomInc. GLM-5.2 jointly generates Chinese and English versions of both the local and global questions, and the two language versions of a question are counted as one QA pair. From this pool, two human reviewers select 1,000 questions for the benchmark without writing or editing any questions or answers.

Training Data Selection and Matching. Excluding the 1,000 benchmark items, the remaining QAs form the training pool. For a fair comparison with ZwZ, for each image, we take the smaller number of available unique questions between the two sides and sample that many from each side without replacement, yielding 8,746 image-matched QA pairs. We then duplicate 254 randomly chosen pairs on both sides to reach 9,000 samples per dataset. Both training sets thus use identical source images (ZwZ retains its bounding-box overlays) and differ solely in the questions.

5 Experiments on ZoomInc Bench

To examine whether questions mined from global-to-zoom perception gaps pose a genuine challenge to current VLMs, we evaluate 28 widely-used VLMs on ZoomInc Bench, spanning closed-source frontier, agentic, fine-grained enhanced single-forward, and open-source general VLMs (Table 2).

Statistics. ZoomInc Bench contains 1,000 questions over 941 images: 472 attribute recognition, 94 counting, and 434 OCR questions. Attribute questions have 4 to 6 answer options; counting and OCR questions are open-ended. Each question has English and Chinese versions (2,000 evaluation instances in total) and an automatically obtained evidence region specified by its crop box. Appendix C shows examples of all three question categories. Table 1 compares ZoomInc Bench with

Table 1: **Comparison with existing fine-grained perception benchmarks.** *QA authoring*: who writes the questions and answers; Model+Human denotes model-generated questions refined by human annotators, and Model denotes questions authored entirely by models, with humans only deciding which candidates to include. *Blind filter*: questions answerable without the image are removed during construction. *Local verif.*: answers are automatically verified on the zoomed evidence region. *Dual-view*: both the full image and the evidence region are provided. *Dedicated distractor generation*: a separate stage for constructing distractor options (Section 3.3). In this column, ✗ indicates that no dedicated stage is reported, and N/A denotes open-ended evaluation. "–" denotes not provided or not specified.

Benchmark	#Questions	QA authoring	Language	Evidence	Blind filter	Local verif.	Dual-view	Dedicated distractor generation
ZoomBench Wei et al. (2026)	845	Model+Human	EN	Auto	✗	✗	✓	✗
V* Bench Wu & Xie (2023)	191	Human	EN	Human	✗	✗	✗	✗
HR-Bench Wang et al. (2025b)	200	Model+Human	EN	✗	✗	✗	✗	✗
VisualProbe Lai et al. (2025)	500	–	EN	✗	✗	✗	✗	N/A
TreeBench Wang et al. (2025a)	405	Model+Human	EN	Human	✗	✗	✗	✗
FINERS-4k Zhang et al. (2026b)	2,427	Human	EN	–	✗	✗	✗	✗
ZoomInc Bench (ours)	1,000	Model	EN/ZH	Auto	✓	✓	✓	✓

Table 2: **Evaluation results on ZoomInc Bench.** Scores are reported as percentages for English and Chinese questions. Each language-specific Overall is weighted by the number of questions in each category. **Bold** and underlined scores indicate the best and second-best results across all models in each column, respectively.

	English				Chinese				Overall
Model	Attr	Count	OCR	Overall	Attr	Count	OCR	Overall	
Closed-source Frontier VLMs									
GPT-5.5 OpenAI (b)	52.75	57.45	60.40	56.51	52.12	63.83	60.96	57.06	56.79
Gemini-3.6-Flash Google DeepMind (2026)	51.27	42.55	**74.71**	60.62	48.52	45.74	**71.42**	58.20	59.41
Claude-Opus-5 Anthropic (2026)	48.09	63.83	71.86	59.89	48.73	62.77	71.29	59.84	59.86
GPT-5.6-Sol OpenAI (c)	**56.57**	62.77	67.93	**62.08**	**54.03**	63.83	65.84	**60.07**	**61.08**
Agentic VLMs									
DeepEyes-7B Zheng et al. (2026)	24.79	20.21	41.05	31.42	24.36	22.34	34.46	28.55	29.99
Thyme-7B Zhang et al. (2025)	26.69	34.04	43.88	34.85	22.88	31.91	40.31	31.30	33.07
DeepEyesV2-7B Hong et al. (2026)	22.25	41.49	43.70	33.37	22.67	34.04	43.62	32.83	33.10
PixelReasoner-7B Su et al. (2026)	27.12	45.74	44.83	36.56	25.42	23.40	40.13	31.62	34.09
Fine-grained Enhanced Single Forward VLMs									
ZwZ-4B Wei et al. (2026)	30.08	39.36	29.68	30.78	33.47	50.00	34.45	35.45	33.12
Vision-OPD-4B Yuan et al. (2026)	35.59	59.57	45.69	42.23	39.62	59.57	43.80	43.31	42.77
Vision-OPD-9B Yuan et al. (2026)	46.40	55.32	47.18	47.58	47.67	60.64	37.97	44.68	46.13
ZwZ-8B Wei et al. (2026)	42.58	55.32	50.53	47.23	44.70	52.13	51.77	48.47	47.85
Open-source General VLMs									
MiMo-VL-7B-RL Xiaomi (2025)	23.73	34.04	20.06	23.11	23.31	41.49	32.06	28.82	25.96
GLM-4.6V (106B) Team et al. (2025)	24.36	38.30	31.57	28.80	22.46	36.17	28.51	26.37	27.59
Qwen3-VL-4B Bai et al. (2025)	21.82	43.62	31.14	27.92	22.88	41.49	33.63	29.30	28.61
Qwen3-VL-8B Bai et al. (2025)	15.89	41.49	39.17	28.40	19.92	30.85	41.64	30.37	29.38
Qwen3.5-4B Team (2026)	21.40	51.06	34.40	29.83	22.67	46.81	34.87	30.23	30.03
Qwen3.5-9B Team (2026)	21.82	41.49	37.79	30.60	22.67	47.87	42.52	33.66	32.13
Qwen3.5-35B-A3B Team (2026)	28.18	44.68	46.10	37.51	27.33	51.06	47.95	38.51	38.01
Qwen3.6-35B-A3B Qwen Team (2026e)	26.91	54.26	50.21	39.59	26.69	52.13	46.17	37.54	38.57
Qwen3-VL-235B-A22B Bai et al. (2025)	29.45	52.13	48.01	39.64	30.93	50.00	50.90	41.39	40.51
Qwen3.5-27B Team (2026)	33.05	51.06	45.92	40.33	34.75	58.51	48.67	43.02	41.68
Qwen3.6-27B Qwen Team (2026b)	33.47	56.38	47.47	41.70	37.71	**70.21**	45.33	44.07	42.89
Qwen3.8-27B Qwen Team (2026c)	40.25	54.26	45.49	43.84	44.92	62.77	43.05	45.78	44.81
Qwen3.5-397B-A17B Team (2026)	33.47	65.96	51.95	44.55	38.14	62.77	50.41	45.78	45.16
GLM-5.3-Flash (320B) GLM-5-Team et al. (2026)	44.07	54.26	43.77	44.90	46.61	55.32	43.74	46.18	45.54
Kimi-K3 (2.8T) Team et al. (2026a)	49.15	64.89	45.48	49.04	46.82	64.89	46.22	48.26	48.65
Qwen3.8-Flash-Next (125B) Qwen Team (2026a;d)	47.46	**67.02**	56.97	53.42	52.97	67.02	58.45	56.67	55.05

existing benchmarks: it is the only one that combines fully model-authored QAs, automated verification, and dedicated distractor construction, and Section 5.2 shows how these distractors reveal errors shared across models.

Evaluation protocol. Multiple-choice questions are scored by exact match on the selected option, and counting questions by exact match on the predicted integer. OCR questions receive their ANLS score if it is at least 0.7, and zero otherwise. Prompts, decoding temperature, and thinking-mode settings for each model are provided in Appendix F.

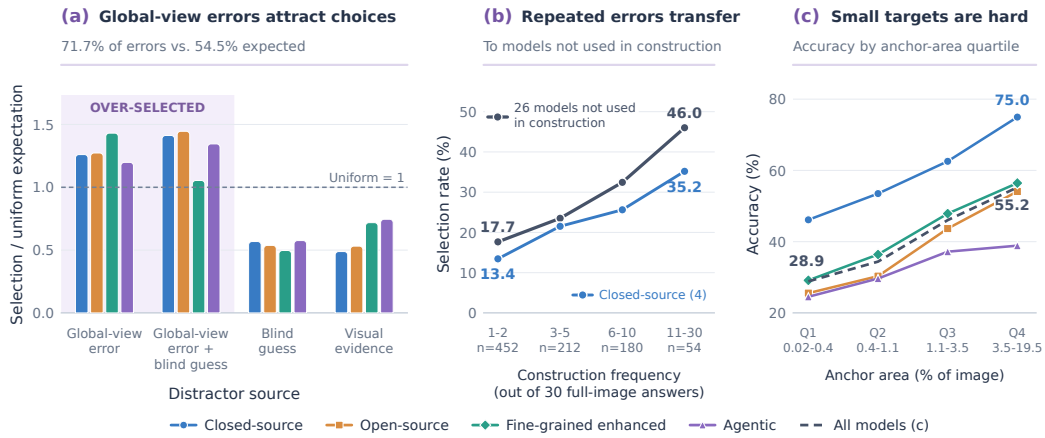

Figure 3: **Where models fail on ZoomInc Bench.** (a) How often each model group chooses each type of distractor compared with the random baseline, considering only incorrect answers. Values above 1 indicate over-selection. *Global-view error + blind guess* denotes distractors produced both as incorrect answers when models view the full image and as guesses based on language priors without an image. (b) How often the 26 models not used in construction (all evaluated models except Qwen3.5-397B-A17B and Qwen3-VL-235B-A22B-Instruct) choose global-view errors, grouped by each error's frequency in the 30 full-image answers collected during construction. (c) Accuracy by quartile of the anchor's area in the full image.

5.1 Benchmark results.

ZoomInc challenges even frontier VLMs. The strongest model, GPT-5.6-Sol, reaches an overall score of only 61.08 (Table 2), and only five models exceed 50. All four closed-source models outperform every open-source general VLM: even GPT-5.5 (56.79) surpasses Qwen3.8-Flash-Next (55.05), the best open-source general model.

Fine-grained enhanced VLMs are highly competitive. ZwZ-8B and Vision-OPD-9B improve over their respective base models, Qwen3-VL-8B and Qwen3.5-9B, by 18.47 and 14.00 points. Despite its smaller size, ZwZ-8B (47.85) also surpasses Qwen3.5-397B-A17B (45.16).

Attribute recognition is the most difficult category. Attribute recognition yields the lowest score averaged over both languages for 23 of 28 models, despite being the only multiple-choice category, with a uniform-guessing accuracy of about 21.5%. Qwen3-VL-8B, for example, falls below this level in both languages (15.89 in English and 19.92 in Chinese), motivating an analysis of which distractors attract incorrect choices.

5.2 Error analysis

Models are misled by global-view errors. Since every distractor retains its source from construction (Section 3.3), we can trace where incorrect choices go. Across all 28 models, global-view-error distractors receive 71.7% of incorrect choices, exceeding the random baseline of 54.5%,[1] whereas blind guesses and visual evidence receive only 0.54 and 0.59 times their random-baseline (Figure 3(a)). This over-selection holds for every model, including those not involved in construction Moreover, wrong answers that appear more often during construction are also chosen more often by other models. Among the 26 models not used in construction, errors that appeared 1–2 times in the 30 full-image answers are chosen 17.7% of the time, versus 46.0% for those that appeared more than 10 times (Figure 3(b)). That is, models reproduce the global-view confusions observed during construction, guessing a scene-plausible answer instead of resolving the queried detail, such as naming a hockey equipment brand for the logo on a hockey player's pants (shown in Figure 8(a) in Appendix E). **In other words, distractors mined from global-view errors capture confusions shared across models, rather than idiosyncrasies of the models used in construction.**

[1]The baseline assumes that whenever a model answers incorrectly, it selects uniformly at random among that question's distractors.

Table 3: **Results of RL training on ZoomInc and ZwZ data.** $\Delta_{\text{ZoomInc}-\text{ZwZ}}$ and $\Delta_{\text{ZoomInc}-\text{Base}}$ denote score gains over the ZwZ-trained and base models, respectively; Avg. is computed over the 13 held-out benchmarks, **excluding ZoomInc Bench**.

	Held-in	Fine-grained Perception							General Perception			Difficult Percept & Reason			
Model	ZoomInc Bench	Zoom Bench	V* Bench	HR 4K	HR 8K	MME-RW Lite	Tree Bench	Visual Probe	CV-Bench	MMStar	MMVP	Baby Vision	Perception Bench	O3-Bench	**Avg.** Only Held-out
Several Large Parameter VLMs															
GPT-5.1	23.22	47.22	70.16	67.00	65.25	45.86	47.65	29.90	84.22	71.60	32.67	13.92	28.97	31.01	48.88
Qwen3-VL-235B-A22B	40.51	49.11	87.96	84.50	81.62	58.16	53.58	46.41	86.72	76.33	64.00	18.30	35.50	60.58	61.75
Qwen3.5-122B-A10B	41.11	51.83	84.82	86.50	83.75	57.89	54.07	44.85	88.06	74.60	70.00	19.07	38.80	65.22	63.04
Qwen3.5-397B-A17B	45.16	54.91	87.43	86.50	85.12	57.06	53.33	49.51	**89.57**	76.00	72.67	20.36	**47.50**	**72.17**	65.55
GLM-5.3-Flash (320B)	45.54	56.80	89.01	87.00	84.75	62.12	**60.25**	48.35	88.94	79.47	74.67	20.10	41.90	65.51	66.07
Gemini-3-Flash	51.27	59.29	86.39	87.88	85.00	55.86	48.64	49.71	**89.57**	**83.60**	**82.67**	**34.51**	43.50	71.01	**67.51**
Our Experimental Models															
Qwen3.5-4B	30.03	48.88	81.68	81.25	77.63	57.89	49.38	43.30	85.99	66.80	57.33	14.69	33.10	54.20	57.86
w/ ZwZ 9K	40.99	57.63	86.91	81.25	79.00	57.43	45.19	52.43	86.58	68.93	53.33	10.82	30.43	56.23	58.94
w/ ZoomInc 9K	**55.81**	59.88	90.05	87.25	83.88	61.59	49.14	**57.28**	88.02	72.53	61.33	21.13	36.23	63.19	63.96
$\Delta_{\text{ZoomInc}-\text{ZwZ}}$	+14.83	+2.25	+3.14	+6.00	+4.88	+4.17	+3.95	+4.85	+1.44	+3.60	+8.00	+10.31	+5.80	+6.96	+5.03
$\Delta_{\text{ZoomInc}-\text{Base}}$	+25.78	+11.00	+8.37	+6.00	+6.25	+3.70	-0.24	+13.98	+2.03	+5.73	+4.00	+6.44	+3.13	+8.99	+6.10
Qwen3.5-9B	32.13	50.30	86.91	82.13	79.63	58.52	49.88	44.66	86.27	71.73	60.00	15.98	34.20	56.52	59.75
w/ ZwZ 9K	44.07	58.93	90.58	85.50	82.38	61.70	47.90	52.62	88.16	73.27	63.33	17.27	34.13	57.10	62.53
w/ ZoomInc 9K	55.48	**60.36**	**93.19**	**90.25**	**85.50**	**63.31**	53.33	56.31	88.58	73.87	70.00	17.53	39.00	68.12	66.10
$\Delta_{\text{ZoomInc}-\text{ZwZ}}$	+11.41	+1.43	+2.62	+4.75	+3.13	+1.62	+5.43	+3.69	+0.42	+0.60	+6.67	+0.26	+4.87	+11.01	+3.58
$\Delta_{\text{ZoomInc}-\text{Base}}$	+23.35	+10.06	+6.28	+8.12	+5.87	+4.79	+3.45	+11.65	+2.31	+2.14	+10.00	+1.55	+4.80	+11.60	+6.35

Smaller targets are harder. Accuracy rises with the fraction of the image occupied by the anchor for every model group (Figure 3(c)): averaged over all models, it increases from 28.9% on the smallest quartile, where anchors occupy less than 0.4% of the image, to 55.2% on the largest.

Appendix D further reports the error analysis regarding open-ended questions, including how incorrect counts differ from the ground truth, as well as common OCR error patterns.

6 Experiments on ZoomInc Training Data

6.1 Main Results

Setup. We conduct the experiment on the image-matched ZoomInc and ZwZ sets described in Section 4, each with 9,000 samples. We select Qwen3.5-4B and Qwen3.5-9B Team (2026) as our base models, applying RL with GRPO Shao et al. (2024) using each set under identical settings (batch size 64, 8 rollouts per question; other details are shown in Appendix G). We evaluate on four groups of benchmarks. *Held-in*: ZoomInc Bench. *Held-out fine-grained perception*: ZoomBench Wei et al. (2026), V* Bench Wu & Xie (2023), HR-Bench 4K and 8K Wang et al. (2025b), MME-RealWorld-Lite Zhang et al. (2024), TreeBench Wang et al. (2025a), and VisualProbe Lai et al. (2025). *General perception*: CV-Bench Tong et al. (2024a), MMStar Chen et al. (2024), and MMVP Tong et al. (2024b). *Difficult perception and reasoning*: BabyVision Chen et al. (2026), PerceptionBench Lin et al. (2026), and O3-Bench Li et al. (2026a). We also compare with some large parameter VLMs, including open-source and closed-source Google DeepMind (2025); OpenAI (a). We report the average score over the 13 held-out benchmarks, excluding ZoomInc Bench. Because ZoomInc Bench and the ZoomInc training data are generated by the same generation pipeline, including it could favor ZoomInc model.

Results. As shown in Table 3, models trained on ZoomInc data consistently outperform their ZwZ-trained counterparts on all 14 benchmarks. Since the two training sets differ solely in the questions, these gains can be attributed to the questions constructed by ZoomInc. Moreover, the improvements brought by ZoomInc extend across all benchmark groups. Compared with the base models, the gains of ZwZ are mainly concentrated on fine-grained perception. In contrast, ZoomInc improves all three groups. The advantage of ZoomInc over ZwZ is most pronounced on difficult perception and reasoning, where the 9B model gains 11.01 points on O3-Bench and the 4B model gains 10.31 points on BabyVision. These results indicate that ZoomInc data enables models to acquire transferable fine-grained perception capabilities.

Remarkably, with only 9K training samples, the 9B model achieves an average score of 66.10, surpassing Qwen3.5-397B-A17B (65.55) and performing on par with GLM-5.3-Flash (66.07), while

Table 4: **Ablation of incremental evidence and dedicated distractors.** Avg. is computed over the 10 held-out benchmarks.

Model	Incremental Evidence	Dedicated Distractors	Held-in	Fine-grained Perception							Difficult Perception & Reason			Avg. Only Held-out
			ZoomInc Bench	Zoom Bench	V* Bench	HR 4K	HR 8K	MME-RW Lite	Tree Bench	Visual Probe	Baby Vision	Perception Bench	O3-Bench	
Qwen3.5-4B	–	–	30.03	48.88	81.68	81.25	77.63	57.89	**49.38**	43.30	14.69	33.10	54.20	54.20
+ZoomInc 9K	✗	✓	49.71	56.45	88.48	85.50	81.75	60.24	48.40	52.82	18.56	34.70	58.84	58.57
+ZoomInc 9K	✓	✗	46.85	57.04	**90.05**	85.75	80.25	**62.69**	48.89	54.76	17.53	34.97	59.42	59.14
+ZoomInc 9K	✓	✓	**55.81**	**59.88**	**90.05**	**87.25**	**83.88**	61.59	49.14	**57.28**	**21.13**	**36.23**	**63.19**	**60.96**

the 4B model (63.96) also surpasses Qwen3.5-122B-A10B and the ZwZ-trained 9B model. Moreover, ZoomInc-trained models achieve the best results among all compared models on 6 of 7 fine-grained perception benchmarks. However, we also notice that the strongest large models still lead on general perception and difficult perception and reasoning, suggesting that these tasks still rely heavily on model scale and reasoning capability.

6.2 Ablation Study

Setup. We ablate two core designs of ZoomInc on Qwen3.5-4B. All variants use 9K training samples, with the same RL settings and evaluation protocol as in Table 3. **(1)** Remove incremental evidence: instead of requiring questions to be based on incremental information that appears only in the zoomed-view caption, we generate questions directly from the zoomed-view caption. The source images, candidate QA pool size, and 9K sample selection procedure are otherwise identical to those of the full version. **(2)** Remove dedicated distractors: we keep each question, its answer, the correct option's position, and the number of options unchanged, and replace only the distractors of each multiple-choice question with correct answers from other training questions in the same answer category (e.g., color or material). OCR and counting questions are open-ended and are unaffected by this ablation.

Results. As shown in Table 4, the results demonstrate that both designs contribute to training performance. The full ZoomInc pipeline achieves the highest average score across the 10 held-out benchmarks. Removing any component incude the performance losses on almost all benchmarks. Both ablations incur the largest losses in the *difficult perception and reasoning* group. The full version improves over the base model by an average of 6.19 points across these three benchmarks, while each ablation retains only about half of this gain. This indicates that the two designs mainly improve the capability of perception and reasoning that transfer beyond the training distribution.

The effect of incremental evidence is most apparent on fine-grained benchmarks: removing it reduces scores on ZoomBench and VisualProbe by 3.43 and 4.46 points, respectively. This indicates that restricting questions to evidence revealed only after zooming forces the model to learn to locate and recognize details hard to discern. Meanwhile, removing dedicated distractors causes the largest drop on the held-in ZoomInc Bench, at 8.96 points. It is expected because the benchmark's options are constructed using the same pipeline. Using dedicated distractors introduces harder confusions, forcing the model to attend more closely to visual details and thereby enhancing its perception capabilities under the same data budget.

7 Conclusion

To address the difficulty of constructing reliable fine-grained visual supervision at scale, we propose ZoomInc, a data construction pipeline that exploits the perception gap between global and zoomed-in views. ZoomInc turns details revealed only by zooming into questions about the full image, verifies their answers on the zoomed view with multiple models, and mines multi-source distractors that could mislead VLMs. With this pipeline, we construct ZoomInc Bench, a challenging benchmark whose questions and answers are entirely model-authored. Through the analysis of failure cases, we find that different VLMs share similar confusions when fine details are unresolved in the full image. Experiments further show the effectiveness of ZoomInc data as training supervision, which consistently improves VLMs' capabilities of fine-grained perception, general perception, and difficult perception and reasoning.

USE OF GENERATIVE AI AND LLM DISCLOSURE

As part of the ZoomInc data construction pipeline, we use LLMs and VLMs for image captioning, anchor discovery and grounding, incremental fact extraction, bilingual QA generation, distractor construction, and quality verification. The pipeline and model assignments are detailed in Sections 3 and 4 and Appendix B. Two human reviewers screen the generated benchmark candidates by accepting or rejecting them, without writing or editing any questions or answers.

During the preparation of this work, the authors use LLMs (**gemini-3.7-flash** and **GPT-5.6-Sol**) strictly for language polishing and grammatical corrections, and assisting in the development of a subset of experimental code. All research ideation, experimental designs, empirical analysis, and conclusions were independently conducted and drawn by the human authors, who authored the primary manuscript. All AI-assisted text and code were meticulously reviewed, edited, and verified line-by-line by the authors to ensure that no hallucinations, erroneous figures, or fabricated citations were introduced. The authors assume full and sole responsibility for the contents of this paper.

REFERENCES

Anthropic. Introducing Claude Opus 5. https://www.anthropic.com/news/claude-opus-5, July 2026.

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. *arXiv preprint arXiv:2511.21631*, 2025.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in Neural Information Processing Systems*, 32: 841–852, 2019.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3558–3568, 2021.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, et al. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087. Curran Associates, Inc., 2024. doi: 10.52202/079017-0850. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2f8ee6a3d766b426d2618e555b5aeb39-Paper-Conference.pdf.

Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. *arXiv preprint arXiv:2203.09065*, 2022.

Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, et al. Meta clip 2: A worldwide scaling recipe. *arXiv preprint arXiv:2507.22062*, 2025.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Paul Gavrikov, Wei Lin, M. Jehanzeb Mirza, Soumya Jahagirdar, Muhammad Huzaifa, Sivan Doveh, Serena Yeung-Levy, James Glass, and Hilde Kuehne. VisualOverload: Probing visual understanding of vlms in really dense scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2026.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Google DeepMind. Gemini 3 Flash Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf, December 2025.

Google DeepMind. Gemini 3.6 Flash Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-6-Flash-Model-Card.pdf, July 2026.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. FaithScore: Fine-grained evaluations of hallucinations in large vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5042–5063, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.290. URL https://aclanthology.org/2024.findings-emnlp.290/.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

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

Kaican Li, Lewei Yao, Jiannan Wu, Tiezheng Yu, Jierun Chen, Haoli Bai, Lu Hou, Lanqing Hong, Wei Zhang, and Nevin L. Zhang. Insight-o3: Empowering multimodal foundation models with generalized visual search. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Pengyu Li, Zhitao Gao, Lingling Zhang, Muye Huang, Yuanming Li, Fangzhi Xu, and Jun Liu. Visual-opsd: Cross-modal on-policy self-distillation for efficient unified multimodal reasoning. *arXiv preprint arXiv:2606.18974*, 2026b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. URL https://aclanthology.org/2023.emnlp-main.20/.

Zichao Lin, Yifeng Xie, Bowen Qu, Haiming Wang, Jia Li, Haoning Wu, Yuhao Dong, Zuhao Yang, Jinguo Zhu, Haoyu Lu, et al. Perceptionbench: Evaluating atomic visual perception in multimodal large language models. *arXiv preprint arXiv:2607.24957*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

OpenAI. GPT-5.1. https://developers.openai.com/api/docs/models/gpt-5.1, a. Accessed: 2026-09-26.

OpenAI. GPT-5.5. https://developers.openai.com/api/docs/models/gpt-5.5, b.

OpenAI. GPT-5.6 Sol. https://developers.openai.com/api/docs/models/gpt-5.6-sol, c.

Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. Technical report, Alibaba Group, August 2026a.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-27b.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL https://qwen.ai/blog?id=qwen3.8.

Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency, August 2026d. URL https://qwen.ai/blog?id=qwen3.8-flash-next.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026e. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4035–4045, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1437. URL https://aclanthology.org/D18-1437/.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6613–6629, 2025.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang,

Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence, 2026a. URL https://arxiv.org/abs/2607.24653.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Yuanying Guo, Xiaoru Hao, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Yue Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Yingwei Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Ao Wang, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Yifei Xin, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Ruihan Yang, Xiaofei Yang, Xinlong Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Wenjie Ye, Zhuorui Ye, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan

Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026b. URL https://arxiv.org/abs/2602.02276.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. URL https://arxiv.org/abs/2507.01006.

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, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024b.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025a.

Wenbin Wang, Yongcheng Jing, Liang Ding, Yingjie Wang, Li Shen, Yong Luo, Bo Du, and Dacheng Tao. Retrieval-augmented perception: High-resolution image perception meets visual rag. In *Forty-second International Conference on Machine Learning*. URL https://arxiv.org/abs/2503.01222.

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

Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, et al. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

LLM-Core-Team Xiaomi. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.736. URL https://aclanthology.org/2025.acl-long.736/.

Kangning Zhang, Yixing Li, Shuai Shao, Qingyao Li, Zhengxi Lu, Zhiyuan Yao, Jianghao Lin, Wenxiang Jiao, Yuan Lu, Weiwen Liu, et al. Vad: Attributing visual evidence for target reconstruction in multimodal on-policy distillation. *arXiv preprint arXiv:2607.28590*, 2026a.

Lu Zhang, Jiazuo Yu, Haomiao Xiong, Ping Hu, Yunzhi Zhuge, Huchuan Lu, and You He. Finers: Fine-grained reasoning and segmentation of small objects with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:67403–67426, 2026b.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images, 2025. URL https://arxiv.org/abs/2508.11630.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, and Chao Shen. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026.

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

A Complete Algorithm and Additional Implementation Details

Section 3 describes the core pipeline. Here we add two implementation details, incorporated into Algorithm 1. Model assignments and thresholds are given in Appendix B; the prompts are collected in Appendix J.

Anchor markers in the zoomed view. The expanded crop V_z^i may contain several objects of the anchor's category. A VLM enumerates these instances, including partially visible ones, and matches the selected anchor to its grounded box. If multiple instances are present, we draw a red rectangle around the anchor itself; if only one is present, the crop remains unmarked. This rectangle is distinct from b_i, which marks the entire crop region in the global view. The marked-view caption prompt uses it only to identify the anchor and otherwise follows the unmarked-view instructions. It must not be described as photographed content or used as an answer. Local questions may refer to the marker; global questions use natural referring expressions. Figure 4 shows both cases.

Joint crop/full-image audit and rewriting. After QA generation, `EDGE_REWRITER` inspects the crop and the full photograph with b_i marked, together with both question forms and the reference answer. It checks whether answer-bearing text or objects continue beyond the crop, whether the global referent is unique, and whether both questions admit the same answer. It returns *keep*, *rewrite*, or *drop*. A bounded rewrite can add answer-free qualifiers valid in both views while preserving the non-OCR answer. For OCR, it may narrow the question and answer to a complete text block or specified prefix/suffix contained in the original answer, but cannot add text read from the full image. Counting questions whose requested set extends outside the crop are discarded. Revisions are synchronized across the local/global and English/Chinese questions, and target and answer constraints are checked again before crop-based verification (Section 3.3). Both anchor enumeration and this audit use Qwen3.5-397B-A17B.



Figure 4: **Conditional anchor markers in the zoomed view.** (a) Two people are present, including the partially visible person on the left; the red rectangle selects the person wearing a red-and-black plaid shawl. (b) Only one person is present, so no marker is added. Both views are rendered from the original images and recorded crop/anchor coordinates using the pipeline's marker and resizing functions. The rectangle identifies the anchor rather than the answer-bearing detail; it is used only in the zoomed view and is not part of the photographed scene.

Algorithm 1 *ZoomInc Data Construction*

Require: Image pool $\mathcal{I}$; configured models; $N = 10$ samples per model; $R = 3$ MCQ verification rounds
Ensure: Verified QA dataset $\mathcal{D}$ with evidence regions and distractor provenance

$\mathcal{D} \leftarrow \emptyset$
for each full image $V_g \in \mathcal{I}$ **do**
  $C_g \leftarrow \textsc{SceneCaption}(V_g)$; $\mathcal{E} \leftarrow \textsc{ProposeAnchors}(C_g)$
  for each referring expression $e_i \in \mathcal{E}$ **do**
  // Anchor discovery and grounding
    $\mathcal{B}_i \leftarrow \textsc{GroundIndependently}(V_g, e_i)$
    if $\neg\textsc{ValidGrounding}(\mathcal{B}_i, 0.5)$ **then continue**
    $b_i \leftarrow \textsc{Expand}(\text{median}_{\text{coord}}(\mathcal{B}_i))$; $V_z^i \leftarrow \textsc{Crop}(V_g, b_i)$
    $L_i \leftarrow \textsc{EnumerateAnchorInstances}(V_z^i, e_i)$
    if no instance matches the grounded anchor **then continue**
    $\widetilde{V}_z^i \leftarrow \textsc{MarkAnchorIfMultiple}(V_z^i, L_i)$
  // Global-to-zoom evidence and QA construction
    $V_g^i \leftarrow \textsc{DrawBox}(V_g, b_i)$
    $C_g^i \leftarrow M_{\text{cap}}(V_g^i, b_i)$; $C_z^i \leftarrow M_{\text{cap}}(\widetilde{V}_z^i)$ // Independent calls
    $\Delta_i \leftarrow \textsc{IncrementalFacts}(C_g^i, C_z^i)$
    $(q_{\text{local}}^i, q_{\text{global}}^i, a_i, t_i) \leftarrow \textsc{GenerateQA}(\Delta_i, e_i, L_i)$
    if no supported, unambiguous QA is generated **then continue**
    $(v, q_{\text{local}}^i, q_{\text{global}}^i, a_i) \leftarrow \textsc{EdgeAuditAndRewrite}(\widetilde{V}_z^i, V_g^i, q_{\text{local}}^i, q_{\text{global}}^i, a_i)$
    if $v =$ drop or post-rewrite constraints fail **then continue**
  // Quality verification and multi-source distractors
    $A_{\text{crop}} \leftarrow \textsc{Sample}(\text{crop}, \widetilde{V}_z^i, q_{\text{local}}^i, N)$
    if $\min_m \textsc{Correct}(A_{\text{crop}}^m, a_i) < 6$ **then continue**
    $A_{\text{blind}} \leftarrow \textsc{Sample}(\text{blind}, \emptyset, q_{\text{global}}^i, N)$
    if $\min_m \textsc{Correct}(A_{\text{blind}}^m, a_i) \geq 3$ **then continue**
    $O_i \leftarrow \emptyset$; $S_i \leftarrow \emptyset$
    if $t_i =$ attribute **then**
      $A_{\text{global}} \leftarrow \textsc{Sample}(\text{global}, V_g, q_{\text{global}}^i, N)$
      $D_v \leftarrow \textsc{VisualAlternatives}(V_g, \widetilde{V}_z^i, q_{\text{local}}^i, q_{\text{global}}^i, a_i)$
      $D_b \leftarrow \textsc{Incorrect}(A_{\text{blind}}, a_i)$; $D_g \leftarrow \textsc{Incorrect}(A_{\text{global}}, a_i)$
      $(O_i, S_i) \leftarrow \textsc{CurateOptions}(a_i, D_v, D_b, D_g)$
      verified $\leftarrow$ false
      for $r = 1, \ldots, R$ **do**
        $A_{\text{mcq}} \leftarrow \textsc{Sample}(\text{mcq}, \widetilde{V}_z^i, (q_{\text{local}}^i, q_{\text{global}}^i, O_i), N)$
        if $\min_m \textsc{Correct}(A_{\text{mcq}}^m, a_i) \geq 6$ **then**
          verified $\leftarrow$ true; **break**
        end if
        if $r < R$ **then** $(O_i, S_i) \leftarrow \textsc{RepairDistractors}(O_i, S_i)$
      end for
      if $\neg$verified **then continue**
    end if
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(V_g, V_z^i, b_i, q_{\text{local}}^i, q_{\text{global}}^i, a_i, t_i, O_i, S_i)\}$
  end for
end for
return $\mathcal{D}$

Notation and operations. L_i records the anchor instances; $\widetilde{V}_z^i$ is the optionally marked zoomed view. EDGEAUDITANDREWRITE returns a verdict v and the retained or revised QA under the constraints above. Both question forms have English and Chinese versions. The remaining operations follow Section 3, with verification settings in Appendix B; S_i preserves distractor provenance during repairs.

B CONSTRUCTION SETTINGS

Table 5 provides the complete model assignments for the ZoomInc pipeline described in Section 4.

Table 5: **Model assignments in the ZoomInc construction pipeline.**

Model	Roles
Qwen3.5-397B-A17B Team (2026)	Initial global scene captioning and paired-view captioning (M_{cap}); grounding; crop anchor enumeration; joint crop/full-image audit and rewriting; blind answering; global-view error collection; visual distractor generation; final MCQ verification.
GLM-5.2 GLM-5-Team et al. (2026)	Incremental fact extraction and QA generation.
Qwen3-VL-235B-A22B-Instruct Bai et al. (2025)	Grounding; crop-based verification; blind answering; global-view error collection; final MCQ verification.
Kimi-K2.6 Team et al. (2026b)	Grounding; crop-based verification; blind answering; global-view error collection; final MCQ verification.
Qwen3.5-122B-A10B Team (2026)	Answer-equivalence judging.
DeepSeek-V4-Pro DeepSeek-AI (2026)	Candidate distractor clustering and option-pool curation.

Verification Thresholds. All verification stages sample 10 answers per model. Crop-based verification requires at least 6 correct answers from each of its two models, Qwen3-VL-235B-A22B-Instruct and Kimi-K2.6. Blind verification rejects a question only when all three models each answer correctly at least 3 times. Final MCQ verification requires at least 6 correct selections from each of its three models.

Verification Rounds and Distractor Repair. Final MCQ verification uses at most three rounds: an initial check followed by up to two further checks after distractor repairs. Each round applies the per-model threshold described above. Candidates that still fail after the final round are discarded.

C BENCHMARK QUESTION CATEGORIES

The three question categories in ZoomInc Bench cover the following fine-grained details. Figures 5–7 show three examples per category, with paired English and Chinese questions. The orange boxes mark the specific details needed to answer each question, with dashed lines connecting them to enlarged views. These boxes are selected for illustration and are distinct from the pipeline's anchor boxes.

Attribute Recognition. Questions concern a target's color, material, shape, object identity, actions, or relations. The examples below ask about a truck's mirror housing, a glass tabletop, and a small logo on a golfer's visor.

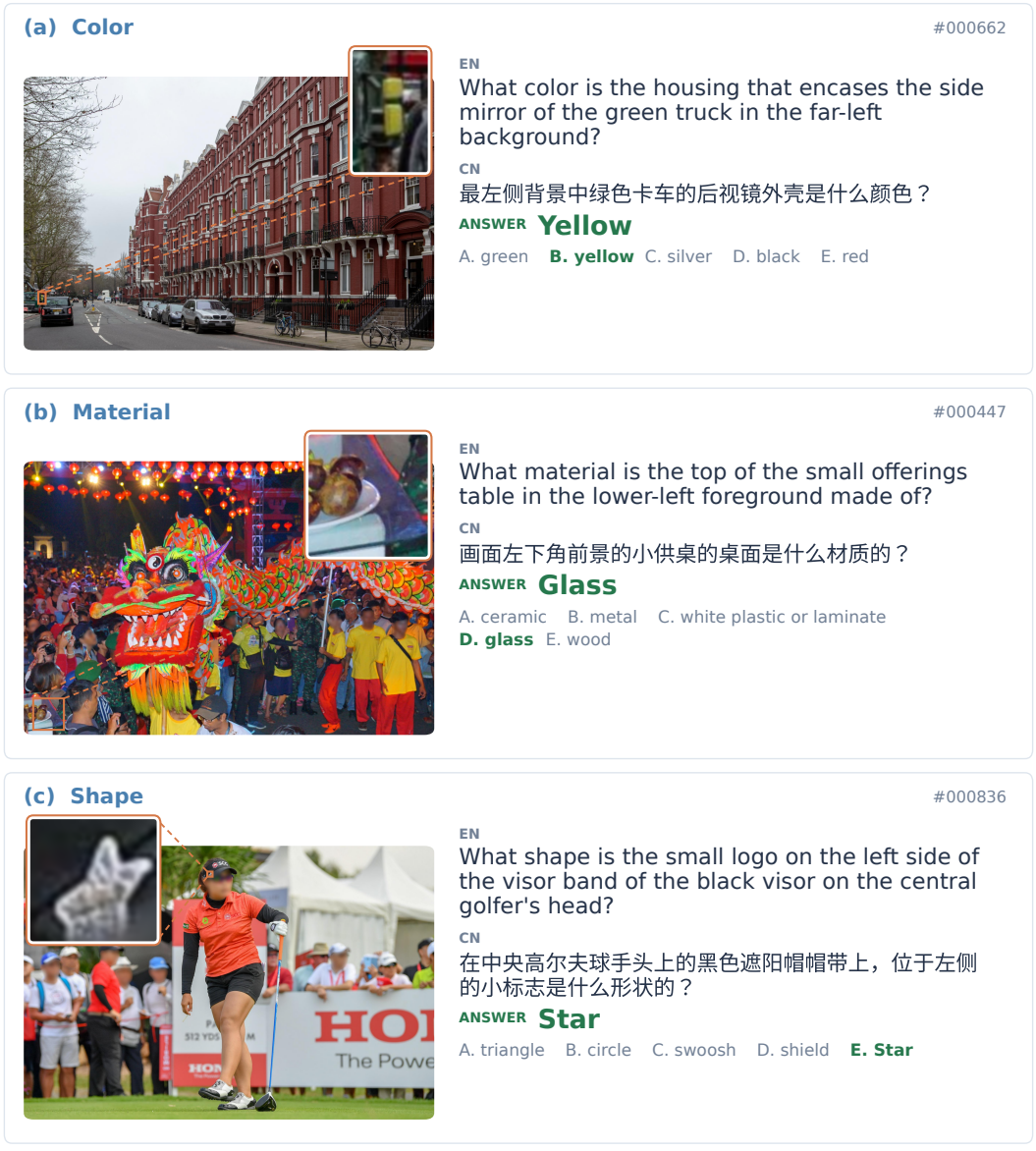

Figure 5: **Attribute recognition examples.** (a) Mirror housing color; (b) tabletop material; (c) logo shape. Correct answers and their corresponding options are highlighted in green.

Counting. Questions ask for the number of entities, object parts, or repeated patterns. The examples below cover repeated images on a menu, people seated on a bench, and pockets on a shirt.



Figure 6: **Counting examples.** (a) Eight burger images on a menu; (b) three people seated inside a shelter; (c) two chest pockets on a shirt. Each question requires an open-ended numerical answer.

Text Recognition (OCR). Questions require transcribing text on signs, vehicles, clothing, printed materials, and product labels. The examples below include a road sign, Japanese lettering on a truck, and Chinese text on a bottle label. Answers retain the original text visible in the image, regardless of the question language.



Figure 7: **Text recognition examples.** (a) A supplementary road sign; (b) lettering on a truck's rear door; (c) the largest Chinese text on a bottle label. Reference transcriptions are shown in green.

D Additional Error Analysis

This section complements Section 5.2.

Per-model consistency. Every evaluated model over-selects global-view errors: they receive 65.0–78.2% of each model's incorrect choices, against 52.3–57.8% expected under uniform choice among the distractors of each question. For the four closed-source models, which took no part in constructing the benchmark, the shares are 75.7% versus 57.4%. Their selection rate of a global-view error also rises with how consistently it arose during construction, from 13.4% when given by one or two of the 30 full-image answers to 35.2% when given more than ten times.

Controlling for option position. Models prefer some option positions over others. Normalizing by the average selection rate of each option position, global-view errors are still selected 1.34 times as often as expected, and 1.46 times when they also coincide with blind guesses, compared with 0.52 and 0.55 times for blind guesses and visual evidence.

Below-chance accuracy of Qwen3-VL-8B. Only 1.5% of Qwen3-VL-8B's answers to attribute questions cannot be parsed, so its below-chance accuracy does not stem from formatting failures. Instead, 67.5% of its incorrect choices fall on global-view errors, against 54.0% expected. Its responses typically claim to inspect the target closely and then state a plausible but incorrect attribute, for example describing a dark blue backpack as black.

Errors on open-ended questions. Counting errors are mostly near misses: 71.6% of incorrect numeric answers are off by exactly one, and 67 of the 94 counting questions have a ground truth of at most three, so the difficulty lies in resolving a few small instances rather than in enumerating many objects. For OCR, among transcriptions without full credit, 29.1% remain close to the reference, with a character-level similarity of at least 0.5; 10.1% omit part of the target text or include adjacent text; and at least 13.4% transcribe other text in the scene instead of the target, as identified by matching predictions against the scene descriptions generated during construction. Most of the remaining transcriptions bear little resemblance to the reference.

E Representative Failure Cases

Figure 8 shows representative failures discussed in Section 5.2.

F ZoomInc Bench Evaluation Settings

All non-agentic models use the same system prompt as ZwZ Wei et al. (2026), which asks the model to reason before placing its final answer within `<answer>` and `</answer>` tags. The user message contains the full image and the question with its options, if applicable. Counting questions additionally request Arabic numerals, while OCR questions request only the transcription within the answer tags, preserving capitalization, spacing, punctuation, and symbols. The Chinese version replaces the question stem while retaining the options and formatting instructions.

Qwen models, ZwZ, and Vision-OPD use a temperature of 0 and a maximum output length of 2,048 tokens, with native thinking disabled via `enable_thinking=false`. MiMo-VL-7B-RL uses the same temperature and output limit, with `/no_think` appended to the question. GLM-4.6V and GLM-5.3-Flash use a temperature of 0 and output limits of 4,096 and 8,192 tokens, respectively; their requests also set `enable_thinking=false`. Kimi-K3 is evaluated with thinking enabled, a temperature of 0, and an output limit of 8,192 tokens. GPT-5.5, GPT-5.6-Sol, Gemini-3.6-Flash, and Claude-Opus-5 use an output limit of 4,096 tokens and retain the provider defaults for temperature and thinking mode.

The four agentic models use their native reasoning and tool-use prompts with multi-turn image interactions, all at a temperature of 0. The per-turn output limits are 10,240 tokens for DeepEyes and PixelReasoner, 20,480 for DeepEyesV2, and 2,048 for Thyme. The total output budget is 32,768 tokens per question. DeepEyes, DeepEyesV2, and PixelReasoner allow up to 11 turns, while Thyme

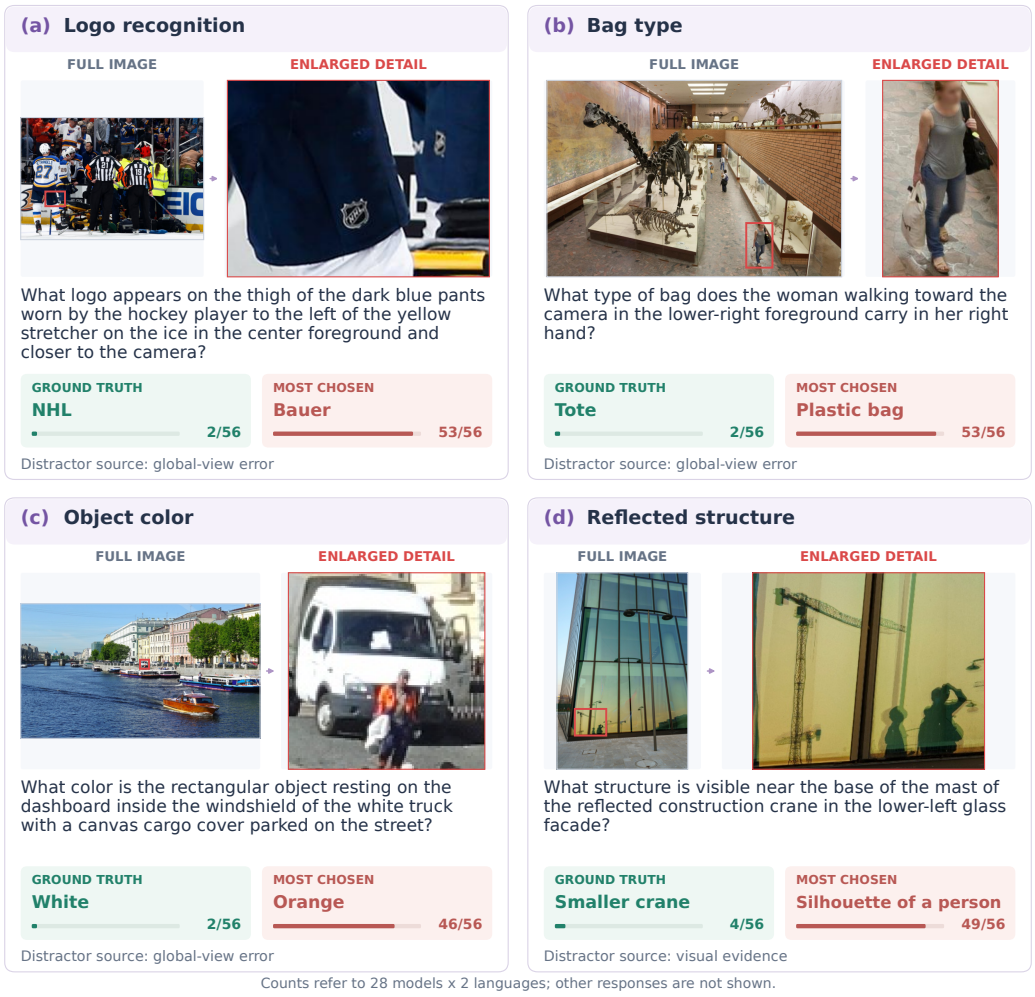

Figure 8: **Representative failures on ZoomInc Bench.** Each case shows the full image with the queried detail outlined in red, an enlarged view of that region, and the responses of the 56 evaluations (28 models $\times$ 2 languages). (a) Nearly all evaluations name Bauer, a hockey equipment brand, instead of the NHL logo, while the blind guesses (Reebok, Nike) are never chosen. (b) The white tote bag is taken for a plastic bag; the visual-evidence distractors, drawn from the black bag on her shoulder and from another pedestrian's purse, are never chosen. (c) Most evaluations (46/56) call the white object on the truck's dashboard orange, likely the color of the vest worn by the worker in front of the truck. (d) The salient reflected silhouettes draw choices away from the smaller crane at the base of the mast.

allows 5; the tool-call limit is 10 for all except PixelReasoner, which allows 4. Generic answer-format instructions that conflict with the native agentic protocols are removed.

G Training and Evaluation Settings

Training. Both datasets are trained with direct RL using GRPO, with a batch size of 64, 8 rollouts per question, and one epoch (140 steps), and the checkpoint at step 140 is evaluated. For both ZwZ and ZoomInc, we use a learning rate of 10^{-6}, no KL regularization, and a maximum response length of 2,046 tokens. ZwZ images retain their native bounding-box overlays, whereas ZoomInc uses the original images. For ZwZ, we follow the official reward definitions Wei et al. (2026). For ZoomInc, rewards follow the ZoomInc Bench scoring rules: multiple-choice and counting questions receive a reward of 1 for an exact match on the selected option or predicted integer, respectively, and 0 otherwise. OCR questions receive their ANLS score as the reward if it is at least 0.7, and 0 otherwise.

Evaluation. For the training experiments, all models are evaluated with chain-of-thought prompting and a temperature of 0. Within each benchmark, base and trained models share the same prompt profile and output limit: 2,048 new tokens for all benchmarks except BabyVision, PerceptionBench, and O3-Bench, which use 8,192 tokens and are scored by GPT-5.5 as the answer judge. MMVP is scored by paired accuracy, CV-Bench by its official balanced score, and MME-RealWorld on 1,919 questions. VisualProbe is reported as the question-weighted average of its easy (141), medium (268), and hard (106) splits.

H GLOBAL-TO-ZOOM EVIDENCE AND QA CONSTRUCTION EXAMPLES

Figures 9–11 illustrate the construction process in Section 3.2 for the three question types. Each example shows the paired views and caption excerpts, selected entries from the original incremental fact list, and how one fact leads to a local QA pair and then a global QA pair. Green text marks details newly described in the zoomed-view caption; purple text highlights the local and global target references. The zoomed views use the original pipeline crops.

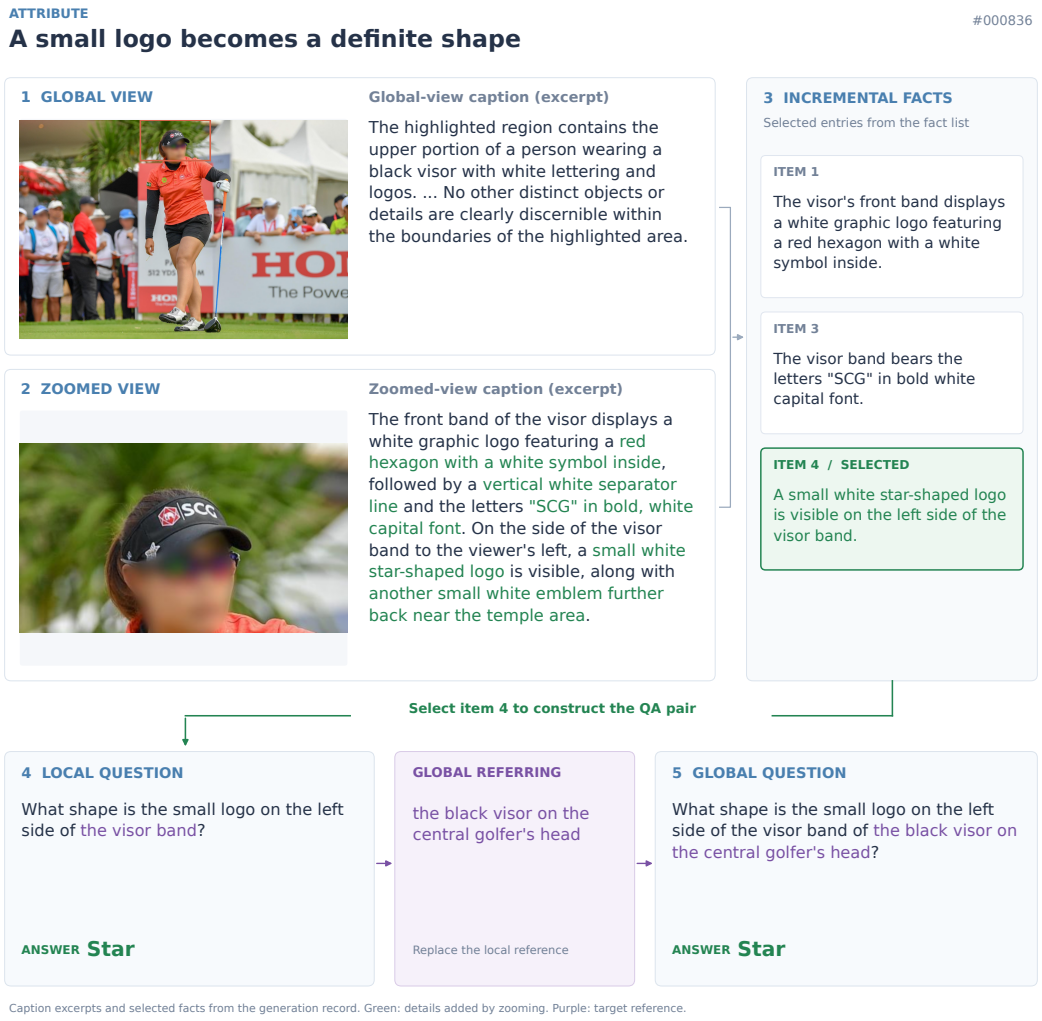

Figure 9: **Attribute QA construction example.** The global-view caption mentions logos on a visor without specifying their shapes. The zoomed-view caption identifies a small star-shaped logo. This fact supports a shape question, which is rewritten using the global referring expression while preserving the answer.

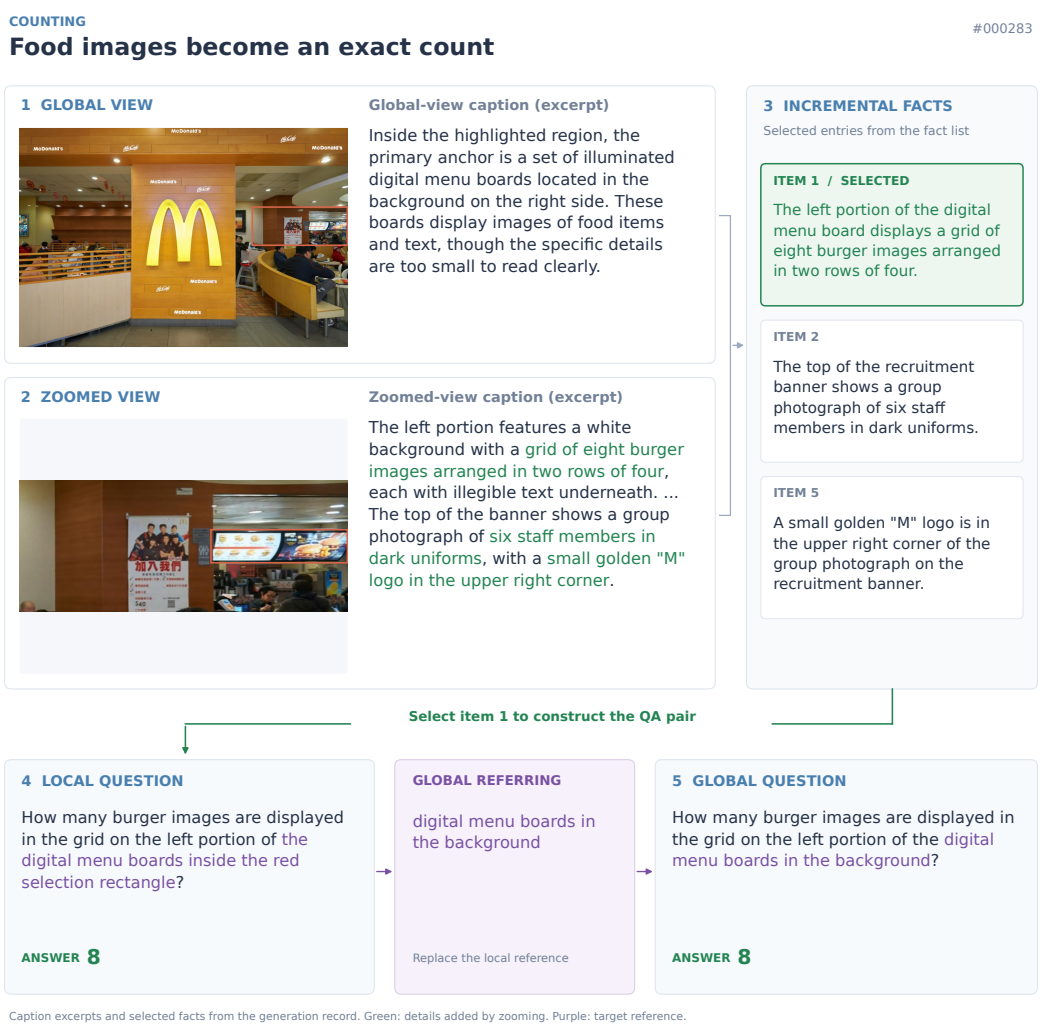

Figure 10: **Counting QA construction example.** The global-view caption describes food images on digital menu boards without giving a count. The zoomed-view caption identifies eight burger images arranged in two rows of four. The selected fact yields a counting question with the same answer in both views.

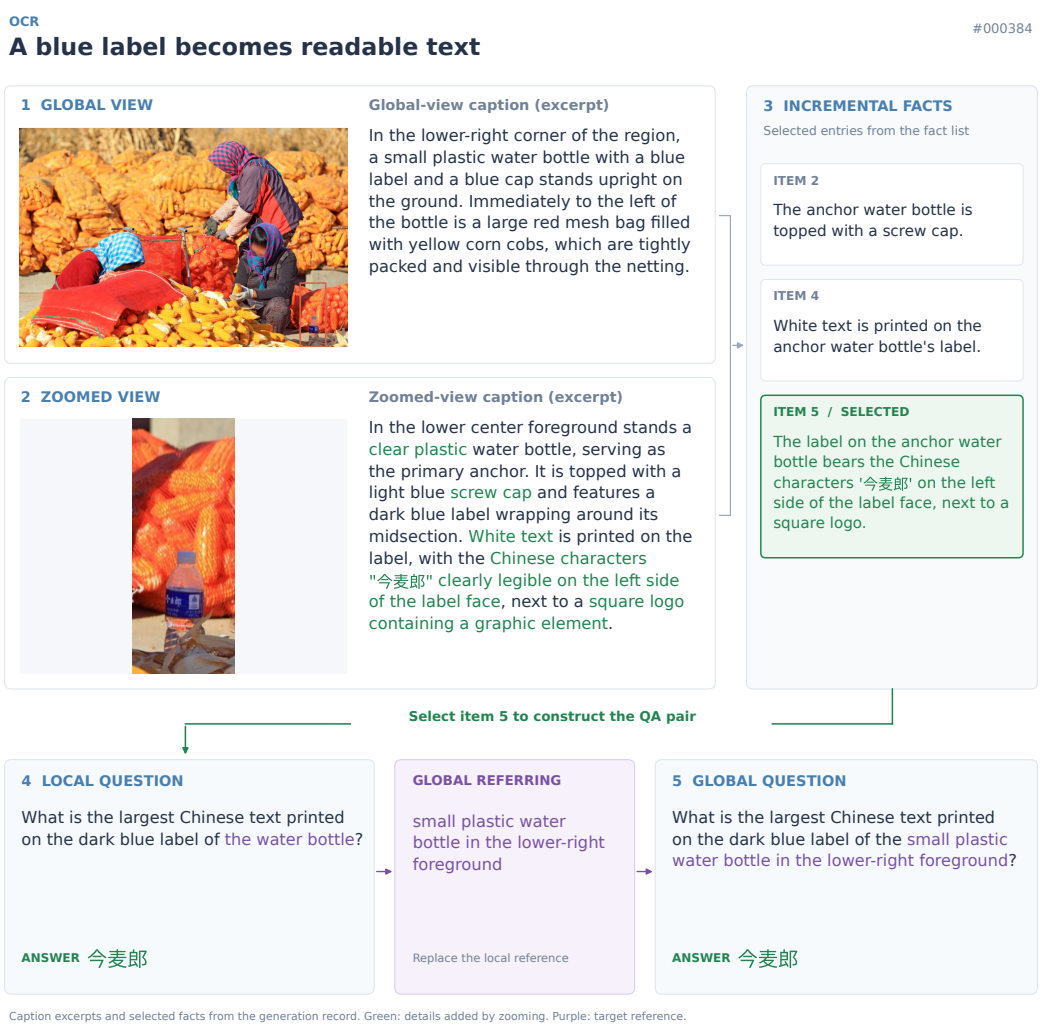

Figure 11: **OCR QA construction example.** The global-view caption describes a bottle with a blue label without transcribing its text. The zoomed-view caption reads the Chinese characters on the label. The selected text fact supports a local OCR question, which is rewritten to identify the bottle in the full image while retaining the original transcription as the answer.

I QUALITY VERIFICATION EXAMPLES

Figures 12 and 13 show candidate QA pairs rejected by the two verification stages in Section 3.3. Crop-based verification checks whether the image supports the generated reference answer, while blind verification filters questions whose answers are predictable without an image. All questions, candidate answers, and vote counts are taken from the original generation records.



Figure 12: **Examples rejected by crop-based verification.** The captions incorrectly describe (A) the race-car digit as "3" instead of "8" and (B) the door handle as silver instead of white. Both verification models give the visually supported answer in all 10 samples, yielding zero matches to the generated reference answer and failing the 6/10 threshold. The original crop inputs have no red anchor markers; orange boxes and enlarged details are added only for display.



Figure 13: **Examples rejected by blind verification.** The reference answers, (A) drumsticks and (B) steam, are supported by the images, but are also predictable from the questions alone. Without images, the three models obtain 30/30 and 29/30 reference matches in total, respectively, with every model meeting the rejection threshold of at least 3/10 matches. The images and blue crop boxes are shown only for the reader; no images are supplied to the blind models.

J PROMPTS USED IN DATA CONSTRUCTION

This section provides the prompts for the roles described in Section 3, plus anchor enumeration, marked/unmarked zoomed-view captioning, and EDGE_REWRITER from Appendix A. Templates are taken from zoom_inc_pipeline/prompts.py; view-dependent instruction blocks are taken from their callers in pipeline.py. Sample-specific placeholders such as {anchor}, {question}, and {reference} are filled at runtime. JSON braces are shown as literal braces rather than Python string escapes. A line of the form <<<INSERT: NAME>>> denotes exact insertion of the named block below, allowing shared instructions to be printed once without omitting their content. The insertion notation is for presentation only and is not sent to the models.

View-specific inputs. CAPTION1 receives the full image with the crop region marked. CAPTION2 receives an unmarked zoomed view, while CAPTION2_ANCHOR_MARKED prepends marker-specific instructions to the same caption template. The local anchor rule in QA generation is selected according to whether the zoomed view has an anchor marker. EDGE_REWRITER receives the crop and the full image with its crop region marked; a candidate box on the crop can additionally locate the referent being audited. Crop solvers answer the local question; full-image and blind solvers answer the global question, with no image supplied for blind answering. The Kimi format instruction is appended to its open-answer requests. Visual distractor generation uses the appropriate crop/full-view rules. MCQ verification sees the crop, local question, global question as context, and assembled choices.

J.1 SHARED INSTRUCTIONS

SPATIAL_FRAME_RULES.

```
SPATIAL FRAME CONTRACT | apply to every direction word you read or
write:
- LEFT/RIGHT BETWEEN OBJECTS uses the viewer's frame of the image: 'the
person to the left of the sign' means left as seen by the viewer. Never
mirror-compute the reference object's own left/right when locating one
object relative to another. When such a phrase is load-bearing, prefer
explicit wording such as 'on the left side of the image'.
- BODY PARTS are anatomical, never viewer-relative: 'right hand' always
means the person's own right hand. Derive the side from the person's
facing direction, never from raw image position: facing the camera ->
the right hand appears on the viewer's LEFT; facing away from the
camera -> the right hand appears on the viewer's RIGHT; facing
image-left -> the left side of the body is toward the camera; facing
image-right -> the right side of the body is toward the camera. If the
facing direction is not clearly visible, never use left/right body-part
wording; identify the part another way (by what it holds or touches, or
with explicit viewer wording such as 'the hand on the left side of the
image').
- FRONT/BEHIND is NEVER a depth word. Scene depth must be written only
as 'closer to the camera (than X)' / 'farther from the camera (than X)'
/ 'beyond X from the camera's viewpoint'. The words front/behind are
reserved exclusively for the intrinsic front/back sides of people,
animals, and vehicles whose facing you have already stated: 'in front
of the seated man' is the side his face and chest point toward, and
'behind him' is his back side, regardless of distance to the camera.
- DEPTH-FLIP TRAP (the most common mistake | check every sentence):
when the camera sees a person's BACK, that person faces AWAY, so
objects deeper in the scene are IN FRONT of that person, and objects
between the camera and the person are behind them. Writing 'behind the
man' for a sign that lies beyond a man seen from the back is WRONG |
write 'in front of the man (farther from the camera)' or 'farther from
the camera than the man'. The same flip applies to seated crowds
watching something away from the camera.
- For objects with no inherent front (poles, boxes, trees, candles,
signs), never use front/behind at all; use camera-distance or
image-position wording.
```

```
- VEHICLE AND BUILDING PARTS use part terms fixed to the object itself:
'the cab end', 'the rear of the trailer', 'the side facing the camera',
'the entrance side'. Never call a vehicle end 'front' unless it is the
actual forward/cab end of that vehicle.
- GROUPS and entities whose facing cannot be determined: use the viewer
frame with explicit image wording ('in the upper-left part of the
image', 'farther from the camera'), never intrinsic terms.
- WHOSE FRAME 'THE IMAGE' MEANS. Image-frame wording | 'in the
upper-left part of the image', 'near the left edge of the image', '图像
左侧' | always refers to the FULL photograph, and to nothing else.
```

CROP_FRAME_CAVEAT.

```
CROP FRAME CAVEAT:
- Your view is a window cut out of the middle of a larger photograph,
so its borders are not the photograph's borders and its left side is
not the photograph's left side. A position read off this view may be
stated only as local to it, never as image-frame wording.
- A FRAME SIDE IS ONLY HONEST NEAR AN EDGE. 'on the left side of the
image' is worth saying about something in the outer third of the
photograph; for anything nearer the middle it reads as a claim the
object is not where it is, and a solver scanning that side will not
find it. When the target sits toward the centre, locate it against a
named object instead | 'just right of the fire hydrant', 'in front of
the bus shelter and a little to its left' | which stays true wherever
the frame is drawn. You cannot see where this crop sits in the
photograph, so when in doubt prefer the named object.
- FRAME SUPERLATIVES REQUIRE THE WHOLE PHOTOGRAPH. leftmost, rightmost,
topmost, bottommost, far left, far right, 最左, 最右, 最上, 最下, 最顶, 最
底 assert that nothing of the kind lies further out, which only a view
of the entire photograph can establish. Without that view do not write
one in either language. Bounded superlatives inside one named host
('the bottom word on the blue sign') are unaffected.
- A HOST'S OWN CORNER NEEDS THAT CORNER IN VIEW. 'the bottom right
corner of the window', 'the top of the sign', '货架最下层' name a
position inside another object, and they are only true if THAT object's
edge on that side is visible here. When the host runs past the border,
the corner you are looking at is the CROP's corner, not the host's, and
the phrase points at something else entirely in the full photograph: a
bookcase cropped to its top two shelves makes 'the book at the bottom
of the bookcase' name a green spine on the second shelf, while the
bookcase's real bottom shelf holds a red one. Use such a phrase only
when you can see where the host ends on that side; otherwise identify
the thing by what it is attached to, what is next to it, or its own
visible attributes.
```

KIMI_FINAL_ANSWER_TAG.

```
Begin your reply with the words The answer is, then put only your final
answer inside <answer></answer>. For example: The answer is
<answer>answer here</answer>.
```

J.2 SCENE CAPTIONING AND ANCHOR GROUNDING

SCOUT_CAPTION.

```
Describe this image thoroughly. Then list every distinct, individually
identifiable object or structure you can see (buildings, boats,
vehicles, people, signs, statues, gates, bridges, etc.). Do not list
background materials such as sky, water, grass, foliage or pavement as
objects.
```

```
Direction wording: give every position in the image frame ('on the left
side of the image', 'in the lower-right foreground'). Express depth
only as foreground / background / closer to or farther from the camera.
Never use 'in front of' or 'behind' a person, animal, or vehicle to
mean nearer to or farther from the camera, and never name a person's
own left/right body side.
```

PHRASE_REWRITE.

```
You are preparing referring expressions for a visual grounding model.

Below is a description of an image. Extract up to {max_phrases}
candidate landmark objects and rewrite each as a SELF-CONTAINED English
referring phrase that would pick out exactly one object instance for
someone looking at the image.

Rules:
- Each phrase must denote exactly ONE object instance. Add a position
word (left / center / right, foreground / background) or a color/size
attribute whenever the bare noun could match several instances.
- Position words inside a phrase must be IMAGE-FRAME words only: left /
center / right, upper / lower, foreground / background, closest to the
camera. NEVER locate a candidate with 'in front of', 'behind', 'ahead
of', or 'beyond' another object or person | depth intent is ambiguous
and grounding models disagree on it. Never use a person's own
left/right body side, and never locate one object through another
object that the description places at a clearly different depth.
- Only concrete, discrete objects: buildings, boats, vehicles, signs,
towers, gates, statues, distinct groups. NEVER: sky, water, grass,
trees, foliage, pavement, shadows, reflections, generic crowds.
- Phrases must rely only on properties visible in the description; do
not invent.
- NEVER use written text, letters, logos or brand names read from the
image as a qualifier (small text is unreliable at low resolution).
- Output strict JSON only:
{"phrases": [{"phrase": "...", "head_noun": "...", "qualifier_type":
"position|color|size|none"}]}

Image description:
{caption}
```

GROUND_NORM1000.

```
Locate the object referred to by: "{phrase}"
Interpret left/right/upper/lower/foreground/background in the phrase as
the viewer's image frame, and foreground/background as distance from
the camera.
Measure x and y on an integer coordinate grid from 0 to 1000 relative
to the full image width and height. Return only one tight bounding box
in this form: [x1, y1, x2, y2]
All four values must be integers between 0 and 1000. Include the
complete visible extent of the referred object. If it is absent or
ambiguous, return []. Do not output any other text.
```

GROUND_NORM01.

```
Locate the object referred to by: "{phrase}"
Interpret left/right/upper/lower/foreground/background in the phrase as
the viewer's image frame, and foreground/background as distance from
the camera.
Measure x and y as decimal coordinates from 0 to 1 relative to the full
image width and height, where 0 is the left/top edge and 1 is the
right/bottom edge.  Return only one tight bounding box wrapped in tags,
in this form:
<bbox>[x1, y1, x2, y2]</bbox>
```

```
All four values must be between 0 and 1. Include the complete visible
extent of the referred object. If it is absent or ambiguous, return
<bbox>[]</bbox>. Do not output any other text.
```

J.3 ANCHOR ENUMERATION AND PAIRED-VIEW CAPTIONING

ANCHOR_INSTANCE_CHECK.

```
Find EVERY distinct visible instance of the concrete object category
"{noun}" in this image crop. Include partially visible instances.

For each instance, draw one tight bounding box around only its visible
extent. Do not infer or include hidden parts. Use integer coordinates
normalized from 0 to 1000 in [x1, y1, x2, y2] order.

Return exactly one JSON object with these fields:
- "count": a non-negative integer equal to the number of detected
instances.
- "instances": a JSON array containing exactly "count" objects.
  Each object must contain:
  - "bbox_2d": four integers [x1, y1, x2, y2].
  - "description": a short, neutral, answer-free visual description.

The following consistency rule is mandatory:
count == len(instances)

If no instance is visible, return exactly:
{"count": 0, "instances": []}

Think carefully before answering. Output JSON only. Do not use Markdown
fences or add explanations.
```

CAPTION1.

```
You are given a full image with one highlighted red rectangular region.
Describe **only the visual content inside the highlighted region and
never mention the rest of the image.** Use the rest of the image only
as context.

The pipeline has pre-selected one anchor object inside this region
(localization metadata only):
"{anchor}"

ANCHOR-CENTERED ORGANIZATION (same convention as the zoomed view):
treat this anchor as the primary reference point of the description,
not as its exclusive scope. First identify the anchor and state every
attribute of it that is discernible at this scale, then cover its
immediate neighborhood, then the rest of the region. Whenever visually
supported, express other objects' positions relative to this anchor.
The anchor phrase is localization metadata only: never copy an
attribute, text, count, or relation from the phrase unless you can
independently see it in the image. If several objects match the phrase,
describe each of them and say the reference is ambiguous; if you cannot
confidently locate the referent, say so and describe the region
normally. The anchor never shrinks your coverage duty: everything else
plainly visible in the region must still be described.
```

```
YOUR CAPTION DEFINES WHAT THE FULL-IMAGE VIEW CAN SEE. A later stage
compares this caption against a zoomed-in description of exactly the
same region and treats every attribute you omitted as "not visible at
full-image scale". The zoomed description follows the same spatial and
inclusion rules as this one; the only intended difference between the
two descriptions is resolution. Omitting something that is in fact
plainly visible therefore corrupts the comparison. Be exhaustive about
everything you can genuinely discern at this scale, and explicit about
what you cannot.

EXHAUSTIVENESS CONTRACT (work from large to small):
- For every large or prominent object in the region | especially people
| state ALL attributes that are plainly discernible at this scale:
garment types and their colors (shirt, pants, dress, hat, shoes),
object colors and materials, obvious pose or action, obviously
identifiable held or attached items, and coarse spatial relations
between the described objects. A large foreground person whose shirt
color is obvious must have that shirt color stated.
- Then cover mid-sized objects the same way.
- For small objects and fine details (small text, small items, precise
counts, subtle patterns, distant faces), report what is discernible and
explicitly say "unclear" or "not readable at this scale" for the rest.
Saying unclear is valuable signal; silently skipping is not.
- Do not compress multiple distinct objects into one vague summary
sentence when their individual attributes are visible.
- When several instances of one category are present (price tags,
signs, windows), bind each described instance to a stable qualifier |
attachment or nearest-object relation, a part location on its host, or
an explicit image-frame position | instead of the bare category noun.

<<<INSERT: SPATIAL_FRAME_RULES>>>
PERSON ORIENTATION: before naming a person's own left/right body side
or a side-specific held item, state which way that person faces and
derive the side per the contract above. When the facing is not
discernible at this scale, use viewer wording ('the hand on the
viewer's left') or say the side is unclear; never guess an anatomical
side.

RECTANGLE-EDGE POLICY (aligned with the zoomed view): the rectangle
boundary is an artificial viewport, not photographed scene content. For
a discrete object that crosses the boundary, apply exactly one branch:
- If it cannot be confidently identified, or no complete part of it
lies inside the region, omit it entirely.
- If it is clearly identifiable and at least one of its parts is
completely visible inside the region, you MAY describe exactly those
fully visible parts and their discernible attributes; say "unclear" for
their small details. Never state whole-object attributes, whole-body
actions, or totals for such an object, and never describe the part
outside the region even though the full image shows it | it is outside
this region's scope.
- Never mention the clipping itself anywhere in the caption: no
"partially visible", "cut off", or "at the edge" wording. Describe what
is inside, say unclear for what is too small, and stay silent about the
rest.
- Large scene surfaces or structures may be described when a
substantial, meaningful portion lies inside the region; do not describe
their truncation.
Before returning the caption, perform a mandatory deletion check:
remove every sentence that mentions the rectangle boundary or an
object's clipped state.
If a small detail is not readable at this scale, explicitly say it is
unclear. Do not guess counts, text, colors, or actions that are not
visually clear.
```

CAPTION2.

```
You are given an image crop constructed around one selected anchor
object.

Selected anchor object (localization metadata only):
"{anchor}"

Write a detailed natural-language caption organized around this
selected anchor and its visible neighborhood. Treat the anchor as the
primary reference point, not as the exclusive scope of the caption.

Coverage and organization:
- First describe the selected anchor itself in detail, including every
clearly visible part, content, marking, readable text, color,
attribute, condition, or action that belongs to it.
- Then describe every other clearly visible object and detail in the
image. Do not omit reliable visual information merely because it is not
part of the anchor.
- Whenever visually supported, express each other object's relationship
to the anchor, preferring frame-free devices: attached to it, contained
in it, on it, nearest to it, or beside it at the same depth. When a
direction word is needed, it must follow the SPATIAL FRAME CONTRACT
below: left/right between objects are the viewer's (write 'to the
viewer's left of the anchor', never the anchor's own left/right),
above/below are image-vertical, front/behind may only name the
intrinsic sides of a person, animal, or vehicle whose facing you have
stated, and any depth difference is written as closer to / farther from
the camera.
- If no reliable relationship to the anchor can be established, still
describe the visible object with a neutral location within this crop.
Never invent a relationship merely to connect it to the anchor.
- DEPTH ZONES (mandatory): this crop is a 2D window that may span very
different scene depths. For every non-anchor object you describe, make
its depth relation to the anchor explicit: in the anchor's own depth
zone, clearly closer to the camera than the anchor, or clearly farther
from the camera. Never say an object is near, beside, or next to the
anchor when it merely appears close in the 2D frame while lying at a
clearly different depth (e.g. a foreground jogger inside a crop built
around a distant lighthouse is 'much closer to the camera than the
lighthouse', never 'near the lighthouse').
- The anchor phrase above is only localization metadata. Do not copy or
assume any attribute, text, identity, or full-scene relationship from
that phrase unless it is independently visible in this image.

<<<INSERT: SPATIAL_FRAME_RULES>>><<<INSERT: CROP_FRAME_CAVEAT>>>
PERSON ORIENTATION AND HANDS (mandatory):
- Before describing any person's pose, clothing sides, held objects, or
body-part sides, state which way that person faces: toward the camera,
away from the camera, toward the image-left, toward the image-right, or
'facing unclear' when it cannot be determined.
- POSE BEFORE SIDE: when two limbs of one person must be told apart,
prefer a pose or state descriptor that needs no laterality at all | the
raised arm versus the arm hanging at the side, the hand gripping the
strap, the gloved hand, the arm resting on the table. Such wording is
unambiguous for any viewer and stays true regardless of facing.
- Name an anatomical side only when a pose descriptor cannot separate
the limbs AND the person's facing is unmistakable at this resolution
(their face, or their back, is plainly readable | not inferred from a
few pixels of hair or shoulder). Then derive the side from that stated
facing per the frame contract and include the viewer-side derivation in
parentheses so it is checkable, for example: 'He faces away from the
camera; his right hand (on the viewer's right) holds a silver can.'
```

```
- If the facing is not unmistakable, never write 'left hand' or 'right
hand': write the pose descriptor, or 'one hand (laterality unclear)'
plus a viewer-frame location. A small or distant person is exactly the
case where a confident side claim is unsafe.
- Whenever you describe what one hand holds, account for the other hand
in the same pass: either describe what it visibly holds (or that it is
visibly empty), or record it in the VISIBILITY AND REFERENCE NOTES
section as out of view with unknown content.

INSTANCE LOCKING FOR REPEATED OBJECTS:
- When the crop shows, or plausibly extends into, more than one
instance of a category | price tags, labels, signs, screens, windows,
wheels, hands, shoes, bags, and similar repeated items | never attach
information to the bare category noun. Bind every described instance to
a stable qualifier that would remain valid in the full image: an
attachment or nearest-object relation ('the mug nearest the kettle'),
or a part location on its host ('on the tail of the aircraft').
- A position in this frame is NOT such a qualifier and never counts as
one. This crop is a window cut out of a larger photograph and its
borders are not the photograph's borders: whatever is leftmost,
rightmost, or 'on the left side' HERE can sit dead centre, or on the
opposite side, of the full image. Never write a frame superlative |
leftmost, rightmost, topmost, bottommost, far left, far right | about
anything in any wording: you cannot see what lies outside this crop, so
you cannot know that anything in it is the outermost one.
- When a rough position genuinely helps a reader picture the scene,
word it as local to this crop ('in the left part of this crop'), never
as a fact about the image. An instance with no attachment or part
qualifier is described without one and recorded in the VISIBILITY AND
REFERENCE NOTES section, not pinned by position.

Include useful details such as objects, readable text, colors,
attributes, actions, and spatial relationships. There is no required
schema, bullet format, or fixed wording; organize the description
naturally.

Base every statement only on visual evidence in this crop:
- HONEST EDGE REPORTING: the four outer crop edges are artificial
viewport boundaries, not photographed scene content. When an object is
cut off by an outer edge (the anchor included), simply say what you
see: describe its clearly visible parts and their attributes, and never
state whole-object attributes, whole-body actions, totals, or overall
shape/size for it, and never guess what continues beyond the edge. Keep
the main description positive | what IS visible | and put every cut-off
/ partially-visible / out-of-view remark in the VISIBILITY AND
REFERENCE NOTES section at the end instead. A truncated object you
cannot even identify belongs only in that section.
```

```
- For text, use a deliberately conservative, character-by-character
standard. A character is clearly legible only when its complete glyph
shape and boundaries are visible and there is no reasonable alternative
reading; prefer a shorter certain transcription over a longer plausible
one, and never use language, brand, or scene context to complete a
word. If the entire inscription meets this standard, transcribe it
exactly. Otherwise never guess the uncertain characters: when a
contiguous substring of at least three characters is clearly legible,
report only that substring, say the rest is unclear, and state where it
sits (its character count and whether it is the beginning, end, or
middle). Call it a beginning or ending only when that text boundary is
itself visible in the crop, and say whether the character touching the
uncertain side has a complete isolated glyph | under cropping,
occlusion, blur, or small scale such boundary characters are
unreliable. If the sign, panel, or surface carrying an inscription is
itself cut off by the crop edge at or near the text region, treat the
inscription as partial with the cut side uncertain even when every
visible glyph is complete | the crop is not a god's-eye view and the
text may continue on the hidden part. Never attribute text to an object
unless it is visibly printed on that object. TEXT COLOR CAUTION:
outlined or two-tone glyphs are easily misread at this scale | dark
characters with a light outline routinely look like white text on a
dark panel. State a text color only when the stroke FILL color is
unambiguous; when uncertain, write 'light-colored text on a dark panel'
or 'dark text on a light panel', or omit the color entirely. A guessed
text color must never appear.
  FIGURE AND GROUND: a panel's color is never the glyphs' color. Name
  the color of the strokes and the color of the surface they sit on as
  two separate facts ('red characters on a yellow panel'), and never
  let one stand in for the other. Watch multi-tone signs especially: a
  fascia whose left panel is green and whose right panel is white has
  TWO panel colors and neither of them is a text color, so writing
  'white letters' for black letters on its white panel is the same
  error as guessing. Illuminated night signage is where this goes wrong
  most.
- COUNTING-SCOPE POLICY: do not state an exact number merely because N
instances happen to be visible in this crop. An exact count is allowed
only in one of these two cases:
  (A) ANCHOR-OWNED SET: the counted items are parts of, attached to,
  worn by, held by, or contained in the selected anchor, and the
  complete anchor-side set relevant to the count is visible.
  (B) FULLY CLOSED ANCHOR-RELATIVE REGION: the counted items lie in a
  finite interval or region defined by the selected anchor and another
  specific visible object, and every boundary needed to delimit that
  region is fully visible in the crop. The crop edges must never serve
  as those boundaries.
  Phrases such as near, around, behind, in front of, to the left/right
  of, in the background, in the water, on the beach, or on the road
  merely locate visible instances; they do not close a counting set.
  Sea, water, beach, road, field, sky, crowd, and other surfaces or
  scene regions that continue beyond the crop are open sets, even when
  every instance currently visible in the crop is sharp and separable.
  For an open set, use non-numeric wording such as 'several are
  visible' when useful, and state that the exact scene total cannot be
  determined from the crop. Never estimate or give a range.
- Do not infer hidden parts or use world knowledge to fill gaps.
Describe uncertain details as unclear.

End the caption with a final section exactly titled 'VISIBILITY AND
REFERENCE NOTES:'. It is a short bullet inventory whose only purpose is
to warn later stages which references would be ambiguous or open; it is
not a description and can never be quoted as visual evidence:
```

```
- List each same-category counterpart whose content is unknown because
it is cut off by a crop edge, out of view, or occluded: the other hand
of a person when you described one hand, additional price
tags/labels/signs beyond the ones you fully described, the unseen
remainder of a vehicle or large object when you described one of its
parts.
- Each note records only: the entity class and its owner or host ('the
man's other hand', 'additional price tags', 'the rest of the blue
trailer'), its status (cut off at which edge, out of view, or
occluded), and 'content unknown' when applicable.
- Notes must NEVER contain attributes, colors, counts, actions, or
transcribed text, and must never guess what the hidden content might
be.
- If nothing qualifies, end with exactly 'VISIBILITY AND REFERENCE
NOTES: none.'
```

CAPTION2_ANCHOR_MARKED.

```
A thin red selection rectangle has been added to this crop because the
crop contains multiple objects of the same category. It marks the one
specific instance named as the selected anchor object in the
instructions below. Use the rectangle silently to distinguish that
photographed object, its parts, contents, markings, and related nearby
objects from those belonging to other same-category objects. Then write
the caption as if the artificial rectangle were absent. Never mention,
describe, count, transcribe, or treat the rectangle, its color, shape,
position, corners, or boundary as photographed content. Refer to the
selected photographed object naturally, for example by its object
category and visible scene position.

<<<INSERT: CAPTION2>>>
```

J.4 INCREMENTAL EVIDENCE AND BILINGUAL QA GENERATION

INCREMENT.

```
Two descriptions of the SAME image region are given. caption1 was
written while seeing the full image (the region was highlighted with a
box). caption2 was written while seeing a high-resolution crop of
exactly that region. Both captions were given the same selected anchor
phrase and were asked to organize around that anchor, so 'the anchor'
in either caption denotes the same photographed object; use that shared
reference to align object identities between them.

Selected anchor object:
"{anchor}"

The phrases 'selected object', 'selected anchor', 'marked object', or
'outlined object' in caption2 refer to this exact photographed anchor,
not to a newly appearing object. Use this metadata only to align object
identity between the two captions. Do not merge a different
same-category object with the selected anchor. If an object's identity
cannot be aligned reliably, omit that candidate fact. Any selection
rectangle, outline, box, marker, label, or annotation mentioned by
caption2 is pipeline metadata, not photographed content, and must never
become an incremental fact.

Extract the incremental facts: discrete, visually verifiable
information that caption2 provides and caption1 either omits (absent),
marks as unclear (unclear), or states differently (contradicted).
Contradictions ARE increments: report them with relation "contradicted"
and record caption1's claim.
```

```
MANDATORY CROP-VIEWPORT INVARIANCE CHECK (higher priority than treating
contradictions as increments): every retained fact must describe the
photographed scene or an object's intrinsic property, and must remain
semantically true if the crop were expanded. The crop is not a
privileged or god's-eye view of the scene; it merely hides content
beyond its four artificial edges. A difference in how much of an object
falls inside the crop is NEVER an absent, unclear, or contradicted
scene fact. Reject any candidate based on an object being at a crop
edge, cut off, cropped, only partially visible, or reduced to
particular visible body parts by the crop. For example, if caption1
describes a person walking while caption2 says only the person's head
and shoulders are visible at the bottom edge, these claims do NOT
contradict each other: caption2 is reporting viewport truncation, so
omit the candidate entirely. Apply this rule even if caption2 states
the boundary observation confidently and even if it can be assigned one
of the allowed fact types. Silently audit every proposed fact for
crop-edge dependence before adding it to the JSON.

VISIBILITY AND REFERENCE NOTES handling: caption2 may end with a
section titled 'VISIBILITY AND REFERENCE NOTES'. That section is
disambiguation context ONLY. Never extract a fact from it, never quote
it in whole or in part as source_from_caption2, and never let it add,
upgrade, or contradict main-description content. Use it solely to stay
aware that other same-category instances or hidden parts exist with
unknown content.

REFERENT QUALIFIER PRESERVATION: when a caption2 evidence sentence pins
an instance with a qualifier | an anatomical side ('in his right
hand'), a stated facing direction, an attachment or nearest-object
relation ('the mug nearest the kettle'), a part location on a host ('on
the tail of the aircraft'), or a depth-zone tag ('much closer to the
camera than the anchor') | the fact's statement MUST keep that
qualifier. Dropping the qualifier and attributing the information to
the bare category ('the man holds a can', 'the price tag reads 140') is
forbidden, because other instances may exist beyond the crop. Never
flip an anatomical side into a viewer side or vice versa when restating
evidence; parenthetical viewer-side derivations from caption2 may be
omitted from the statement, but the anatomical side itself must be
preserved exactly.

SPATIAL-FRAME CONFLICTS ARE NOT INCREMENTS: both captions follow the
same frame contract (anatomical body sides derived from stated facing;
viewer frame between objects; no depth 'behind'). If the two captions
disagree ONLY about which body side of the same person carries an item
or pose (left vs right hand/arm/leg/shoulder), and caption2 states the
person's facing with a derivation while caption1 does not, treat
caption2's side as a corrected relabeling of the same observation: do
NOT emit the side flip itself as a contradicted fact, and never create
a fact whose main content is which body side something is on. Silently
use caption2's side inside any other fact about that item.

CAPTION1 OMISSION GUARD (applies to relation 'absent' only): caption1
is written at full-image scale but may still simply forget an obvious
attribute. When caption1 clearly describes an object as large or
prominent within the region | for example a person occupying much of it
| and the candidate fact states a coarse attribute any full-image
viewer would plainly see (dominant clothing color or garment type,
overall object color, coarse pose, obvious ongoing action), treat the
missing attribute as a caption1 omission, NOT an increment, and omit
the fact. Keep 'absent' facts only for genuinely fine-grained
information: small text, small parts or accessories, precise counts,
subtle patterns, small held items, or details whose visibility
plausibly requires zooming in. This guard never applies to facts whose
relation is 'unclear' (caption1 explicitly could not see it) or
'contradicted' (caption1 made a different claim).
```

```
Only keep facts of these types: count, OCR text, color, action, local
attribute, spatial relation. Discard:
- subjective wording and anything hedged with "possibly/maybe";
- texture-only remarks (foliage, grass, sky, water, lighting);
- facts about the region boundary itself (what is inside/outside the
box, objects cut by the crop edge);
- range counts ("15-20") and estimates;
- hue-level color differences (red vs orange, cyan vs blue) | keep a
color contradiction only if the two claims are clearly different color
classes.

COUNT FACT ELIGIBILITY (mandatory and higher priority than
contradiction):
- Retain an exact count only when it belongs to exactly one of two
scope types: (A) anchor_owned | items that are parts of, attached to,
worn by, held by, or contained in the selected anchor; or (B)
closed_anchor_relative_region | items inside a finite region explicitly
bounded by the selected anchor and another specific object, with all
boundaries fully visible in caption2.
- The artificial crop boundary never closes a set. Relative words such
as near, around, behind, in front of, left/right of, in the background,
in water, or on a beach/road/field are locators, not complete
boundaries. A visible subset of people or objects on a sea, beach,
road, field, sky, crowd, or other region that may continue beyond the
crop is not an exact scene count. Reject it even if caption2 gives a
confident number and even if caption1 gives a different number.
- For every retained count fact, add count_scope_type (anchor_owned or
closed_anchor_relative_region), counted_entity,
scope_boundary_description, and scope_fully_visible=true. If you cannot
truthfully fill all four fields from caption2, omit the count fact. For
non-count facts, use null for these fields.

OCR evidence rules:
- TYPE DISCIPLINE: any fact whose statement reports readable
characters, letters, digits, words, logo text, or an inscription MUST
use type 'OCR text' with all audit fields filled. Never type a
text-reading fact as attribute, local attribute, or color | doing so
would smuggle it past every OCR boundary rule.
- Preserve caption2's exact certainty boundary. If caption2 says only a
substring is legible, label the fact as partial OCR text and preserve
the exact substring, its character count, and its stated position.
- A partial OCR fact is eligible only when it contains at least three
contiguous clearly legible characters and caption2 identifies a usable
position for that substring.
- Never fill in, repair, normalize, or guess unclear characters. Never
turn a partial substring into a claim about the full inscription, word,
brand, or name.
- If caption2 transcribes a sequence but also says the inscription
extends into a cropped or obscured side, treat it as partial OCR even
when the visible sequence itself is described as clear. Preserve which
side is uncertain and whether the boundary-adjacent glyph is fully
isolated.
- For every OCR fact, add these audit fields: ocr_scope (full or
partial), visible_sequence (the exact transcribed characters),
uncertain_side (left, right, top, bottom, both, internal, none, or
unknown), and boundary_character_isolated (a JSON boolean true or
false, never a quoted string). If any portion of the same inscription,
sign, label, or printed text is described as cropped, obscured,
incomplete, or unclear, ocr_scope MUST be partial. Set
boundary_character_isolated to true only when caption2 explicitly
confirms that the glyph touching the uncertain side is complete and
visibly separated on both sides; otherwise use false.
```

```
- For every OCR fact, also express the evidence relative to NORMAL
READING ORDER rather than only to image geometry. Set
substring_position to full when the complete inscription is visible,
prefix when the visible sequence is the beginning of a longer
inscription, suffix when it is the ending, or unknown when caption2
does not support that decision. Set unsafe_boundary to beginning or end
when the corresponding boundary character of visible_sequence touches
an uncertain region; use both, internal, none, or unknown when
appropriate. Do not trim visible_sequence yourself. Physical
top/bottom/left/right and sequence beginning/end are different
concepts.

Output strict JSON only:
{"incremental_facts": [{"type": "...", "statement": "...",
"source_from_caption2": "...", "ocr_scope": "full|partial|null",
"visible_sequence": "...|null", "uncertain_side":
"left|right|top|bottom|both|internal|none|unknown|null",
"substring_position": "full|prefix|suffix|unknown|null",
"unsafe_boundary": "beginning|end|both|internal|none|unknown|null",
"boundary_character_isolated": false, "count_scope_type":
"anchor_owned|closed_anchor_relative_region|null", "counted_entity":
"...|null", "scope_boundary_description": "...|null",
"scope_fully_visible": true, "relation_to_caption1":
"absent|unclear|contradicted", "caption1_claim": null}], "skip_reason":
null}
If there is no reliable increment | in particular if every candidate
fact concerns only vegetation, sky, water, lighting, or generic texture
| return an empty list with a short skip_reason.

caption1:
{caption1}

caption2:
{caption2}
```

QA_GENERATION.

```
You are generating fine-grained visual QA from incremental facts.

The anchor object (salient and uniquely locatable in the full image)
is:
"{anchor}"

Crop-local anchor rule:
{local_anchor_rule}

Incremental facts (information visible only when zooming into the
region around the anchor; each has an evidence sentence from the
zoomed-in caption):
{facts}

Full zoomed-view description of the region (use it to check for
duplicate objects of the same category | NOT as a source of new facts):
{caption2}

Current generation target plus feedback from prior QA candidates for
this SAME anchor:
{retry_context}

Generate up to {qa_per_anchor} questions.

Every incremental fact has a source_fact_index. Return that exact
integer for the one fact used by each QA. Copy evidence_sentence
directly from that fact's source_from_caption2 when available;
otherwise use its statement.
```

```
Question rules:
- Each answer must come from exactly one incremental fact.
- When generating more than one QA in the same response, every QA must
use a different source_fact_index and evidence sentence; never make two
questions from the same fact or paraphrase the same question.
- Any question that asks the solver to read or transcribe text,
letters, characters, digits, words, logo lettering, or a specified
character span MUST use question_type=ocr. Never label such a question
as attribute, comparison, or another type.
- PRIORITY: if one or more eligible OCR facts are present, generate QA
from those OCR facts before using count, attribute, action, or spatial
facts. The incremental-facts list is ordered with OCR facts first.
- Read the ENTIRE OCR fact, especially statement and
source_from_caption2; do not treat visible_sequence as the only useful
information. Visual details explicitly tied to the text|such as its
background color, text color, position, orientation, containing panel,
or relation to another label|are valuable referring cues. Prefer a
question that uses those cues to identify one text region over a
generic question that asks for every word on a whole sign or logo. Use
a visual attribute as a locator, not as another requested answer.
- A HEDGED IDENTITY IS NOT A REFERENT. When caption2 hedges what
something is | 'what looks like a bird', 'possibly a helmet', '某种容器'
| it is telling you it could not identify the object, and a referring
expression built on that guess matches nothing when the guess is wrong:
a solver hunting for a bird will never find the plastic bag it actually
is. Drop the hedged noun and locate the thing by what IS certain |
where it sits, what it is inside or attached to, its color and size
('the small white object on the ledge above the second window'). If
nothing certain identifies it, skip the fact. Never carry the hedge
into the question either: 'the shoe or boot' tells a solver you are
guessing.
- LOCATOR ATTRIBUTE RELIABILITY: prefer host-level cues | panel/sign
color, position, containing object, nearest landmark | over the text's
OWN color. Small text color is frequently misperceived (dark outlined
glyphs read as white), and a wrong locator attribute misleads solvers
or matches a different text region. Include the text's color only when
the fact states it without hedging AND it adds discrimination that the
host cue does not already provide; when the panel alone is unique,
leave the text color out. Never take a color from one region of a
multi-tone sign and attach it to text sitting on another region; if you
cannot say the stroke fill without hedging, use the host cue alone.
- ONE HOST PER OCR ANSWER. The answer is text carried by a SINGLE
bounded host | one sign, label, plate, panel, or printed line. Never
assemble an answer out of characters sitting on separate objects, and
never scope an OCR question by nearness to a GROUP ('the digits printed
near the green bottles', 'the letters around the potted plants'):
proximity to a set has no boundary a solver can reproduce, so which
characters belong in the answer becomes arbitrary and a second reader
will include a different set. If the text you want spans separate
hosts, ask about one of them instead.
- OCR facts may contain deterministic fields computed by code:
safe_sequence, safe_character_count, required_span, and qa_eligible.
Never correct, repair, or invent characters beyond safe_sequence. The
answer must be one clearly supported CONTIGUOUS segment of
safe_sequence.
- For a partial OCR fact (required_span=first or last), write a PLAIN
text-reading question about that inscription, located by host-level
cues, with the answer equal to safe_sequence verbatim. Do not add any
span wording (no 'first/last k characters'), do not add totality words
such as 'entire', 'complete', or 'all the text', and never mention
cropping or hidden characters. A dedicated edge-rewriting stage that
sees the full photograph will rewrite such drafts into the final safe
span wording; your draft only needs to pick the right inscription and
copy safe_sequence.
```

```
- LOCALIZED FULL-OCR SEGMENT: when required_span=full and the fact
itself explicitly associates different complete words or consecutive
word groups with independently identifiable visual regions, you may ask
for one such region and use only its complete text as the answer. The
chosen answer must be one or more WHOLE consecutive words appearing
verbatim in safe_sequence; never take characters from inside a word.
Both questions must include the fact-supported visual locator that
isolates that segment. Do not invent a color, position, container, or
relation. If the fact does not support a unique localized segment, use
the complete safe_sequence instead.
- Example: if a fact says that OPEN appears in white on an upper green
panel and 24 HOURS appears on a separate lower black strip, prefer
asking what text appears on the upper green panel (answer: OPEN),
rather than what all text on the sign says. This example describes the
policy only; never copy its objects, colors, positions, or text into
another image's QA.
- For required_span=full, either ask for the complete clearly legible
text and use safe_sequence verbatim, or apply LOCALIZED FULL-OCR
SEGMENT exactly as defined above.
- Never write view-dependent instructions such as 'excluding the top
character', 'ignore the cropped character', 'remove the first visible
letter', or any reference to an image/crop boundary, cropping,
truncation, or hidden characters. A fixed first/last k span must have
the same answer in the full image and zoomed image.
- Skip every OCR fact with qa_eligible=false or without all required
deterministic fields.
- If the context contains prior-candidate entries in addition to the
generation_target_state, choose a DIFFERENT incremental fact and answer
property. Do not repeat or merely paraphrase any rejected or retained
global/crop question, evidence sentence, or source fact.
- Use rejection feedback only to choose a better-supported candidate.
NEVER alter the reference answer to imitate verifier outputs, and never
invent a fact absent from the incremental-facts list.
- If a previous candidate failed crop verification, prefer a different
fact whose answer is larger/clearer and directly visible in the crop.
- The generation_target_state entry in the context is authoritative. If
zoom_required_still_needed=true and a previous accepted candidate was
answerable from the full image, prefer a substantially smaller or finer
detail marked absent/unclear/contradicted. If it is false, a retained
full-image-answerable QA counts toward the accepted-question minimum;
the next distinct accepted QA may be either full-image-answerable or
zoom-required.
- If retry feedback says a previous candidate remained ambiguous AFTER
the dedicated image-aware referent repair failed, use a different
uniquely referable target. Never add answer-leaking qualifiers.
- DISAMBIGUATION: before writing each question, scan the full
description above. If the answer object's category could match more
than one object in the crop (e.g. a ring worn by a person vs a
standalone ring product image), the crop-local referring phrase MUST
include a qualifier that separates them (position, size,
displayed-vs-worn, color). If no crop-visible qualifier can make it
unique, skip that fact.
- Write TWO semantically paired versions of every question:
  1. question_zh / question_en are GLOBAL questions for the original
  full image. They MUST make the answer object locatable in the full
  image by a solver who has never seen the crop. Choose the locator by
  the target's REAL relation to the anchor (see GLOBAL LOCATOR POLICY
  below).
  2. crop_question_zh / crop_question_en are LOCAL questions for the
  exact zoomed crop. Remove context that the crop itself cannot
  establish, such as "in the center of the plaza", "on the building in
  the background", or "to the right of the white building". Keep only
  qualifiers whose evidence is visible inside the crop and that are
  needed to identify the answer object there.
```

```
- Follow the Crop-local anchor rule exactly. A selection rectangle may
be used only as a referring-expression locator for the photographed
anchor, its parts, contents, markings, or related nearby objects. Never
ask about the rectangle itself, its color, shape, boundary, corners, or
any artificial label. The GLOBAL questions must never mention a
selection rectangle.
- The GLOBAL and LOCAL versions MUST ask the same property of the same
object and MUST have the same answer. The local version is not a new
question.
- The LOCAL question must be self-contained and its answer object must
be uniquely identifiable inside the crop. If removing unavailable
global context makes that impossible, skip the fact.

GLOBAL LOCATOR POLICY (the crop is a 2D window; objects inside it can
lie at completely different scene depths, so 2D adjacency inside the
crop is NOT scene proximity):
  (a) If the answer object has a REAL, visually supported scene
  relation to the anchor | it is a part of it, attached to it, on the
  same supporting surface, held by it, or genuinely adjacent at the
  same depth | locate it through the anchor phrase as usual.
  (b) If the answer object is clearly at a different depth than the
  anchor (the zoomed description says it is much closer to or farther
  from the camera, e.g. a foreground jogger inside a crop built around
  a background lighthouse), NEVER claim proximity with words such as
  near, beside, next to, 附近, 旁边, and never use front/behind as a
  disguised depth word. That fabricates a relation that does not exist,
  misleads the solver into searching around the anchor, and reads
  unnaturally. PREFER KEEPING THE ANCHOR in the question through an
  HONEST relation | it is the verified, easiest-to-find landmark |
  choosing whichever of these reads naturally:
      - CAMERA-AXIS relation (works for ANY anchor, including faceless
      structures; preferred when the target lies on or near the
      anchor's line of sight, which crop co-membership usually
      guarantees): state the depth order along the viewing axis with
      the canonical wording | target closer to the camera: 'the yellow
      taxi between the clock tower and the camera' / 在钟楼与镜头之间的黄色
      出租车; target farther: 'the brick warehouse beyond the red barge,
      farther from the camera' / 越过红色驳船更远处的砖砌仓库. When the two
      do not overlap in the image, add a small image-frame offset word
      ('below the clock tower and between it and the camera'). This
      wording is honest 3D geometry, never a proximity claim.
      - INTRINSIC ANCHOR-FRAME relation: when the anchor is a person,
      animal, or vehicle whose facing caption2 states, the target may
      be located in the anchor's own frame with each side derived
      mechanically per the contract. Example: a cyclist riding toward
      the viewer has the deeper scene BEHIND him and his left-hand side
      on the viewer's right, so 'the bright blue vertical sign behind
      the cyclist on his left-hand side' is honest and natural.
      SIDE-BLINDNESS GUARDRAIL: verification models cannot reliably
      resolve a person's left/right sides, so any anchor-frame
      directional clause is narrative scaffolding only | the question
      MUST still identify the target uniquely when that clause is
      ignored, through at least one anchor-independent discriminator
      (object type plus color, nearest fixed landmark, or viewer-frame
      position). If uniqueness would depend on the anchor-frame clause
      itself, do not use this option.
      - EXPLICIT IMAGE-FRAME relation to the anchor: honest 2D wording
      such as 'in the image, above the cyclist in the background' or
      'below the lighthouse and much closer to the camera', never
      wording that implies physical proximity.
```

```
      - SELF-LOCATING referent (fallback when no anchor-linked wording
      reads naturally, e.g. the linkage would feel contrived across a
      large cluttered depth gap): drop the anchor and locate the target
      with full-image-frame position plus its own visible attributes,
      e.g. 'the jogger in the lower-center foreground of the image
      wearing a cap'. Depth words like foreground/closer to the camera
      stay true across views; horizontal words must be re-derived from
      the FULL image, never copied from the crop question.
  (c) If none of these produces a referent you are confident is unique
  in the FULL image, skip the fact.
  Prefer facts about the anchor itself or objects in its own depth
  zone: a crop-foreground object much closer to the camera is usually
  large and legible in the full image, so its facts rarely survive the
  full-image difficulty gate; spend them only when nothing anchored
  remains.

- Example pair:
  GLOBAL: "What text is written on the sign in the center of the
  plaza?"
  LOCAL: "What text is written on the sign?"
- NEVER put the answer attribute itself into either referring
expression (if the answer is a color, the referring phrase must not
mention that color).
- The answer must be a single definite value: an exact count, one color
word, a short text string, or a short action. Skip facts with ranges,
hedges, or subjective wording.
- The answer itself must never be an anatomical body side ('left hand',
'right leg') NOR a part location on an oriented object ('near the front
left corner', 'at the rear end', 车头/车尾): all vision models share
systematic mirror and orientation confusion, so such answers cannot be
verified reliably. Sides and part locations serve only as referring
qualifiers inside questions.
- COUNTING SOURCE GATE: before using a count fact, classify its
real-scene scope. It is eligible only as (A) anchor_owned: the counted
items are parts of, attached to, worn by, held by, or contained in the
selected anchor, with the complete relevant anchor-side set visible; or
(B) closed_anchor_relative_region: the items occupy a finite interval
or region explicitly bounded by the anchor and another specific visible
object, and all boundaries are fully visible. If neither classification
is supported, skip the fact, even when it contains an exact number and
is contradicted by caption1.
- Crop edges never close a counting scope. Phrases such as near,
around, behind, in front of, left/right of, in the background, in
water, or on a beach/road/field identify a visible subset but do not
define a complete set. Never generate a question asking for the total
of such an open scene region. In particular, counting the instances
currently visible in a cropped slice of sea, beach, road, crowd, field,
or sky is forbidden.
- For an eligible anchor_owned count, the question must explicitly bind
the counted items to the anchor. For an eligible
closed_anchor_relative_region count, the question must name both
bounding objects and the finite interval/region between them. A generic
phrase such as 'behind the anchor' is insufficient.
- Prefer count facts carrying count_scope_type and
scope_fully_visible=true. Independently audit the evidence and full
crop description; metadata cannot rescue an actually open scope.
- Skip facts about the region boundary and vegetation/sky texture.
- Skip color facts that differ only at hue level between the two views.

<<<INSERT: SPATIAL_FRAME_RULES>>><<<INSERT: CROP_FRAME_CAVEAT>>>
```

```
- Apply the frame contract above to every direction word in questions
and answers. Body-part sides stay anatomical exactly as the fact states
them ('right hand' means his own right hand); never convert an
anatomical side into an image side or vice versa. Keep parenthetical
facing/viewer-side derivations OUT of the question text | questions
carry only natural human phrasing.
- BILINGUAL FRAME PARITY: the Chinese and English versions must encode
the same frame with the same discipline. In Chinese, scene depth must
be written as 背景中 / 前景中 / 离镜头更远 / 更靠近镜头 | NEVER as 后方、身
后、前方 or 前面 attached to a person or animal (write 跑者背景中更远处的车
辆, not 跑者后方的车辆). Use 身后/面前 only for a person's intrinsic sides
and only when the question itself states that person's facing (e.g. 背对
镜头的人身后). 左手/右手 are anatomical only, exactly like left/right hand
in English.
- LIMB QUALIFIER PREFERENCE: to single out one of a person's paired
limbs, prefer the pose or state the fact records | the raised arm, the
arm hanging at the side, the hand holding the strap | over an
anatomical side. Pose wording is natural, needs no facing derivation,
and is the only kind a solver can check; verification models do not
reliably resolve a person's left from their right. Use 'left/right
hand' only when the fact states the side with an explicit facing
derivation AND no pose descriptor separates the limbs.
- REFERENT CLOSURE (mandatory): if the fact's answer object belongs to
a multi-instance category | hands, feet, shoes, price tags, labels,
signs, screens, wheels, windows, bags, or similar repeated parts and
products | and the zoomed description or its VISIBILITY AND REFERENCE
NOTES indicate that other instances exist anywhere in the scene, are
cut off, or are out of view with unknown content, then BOTH the global
and crop questions MUST pin the exact instance using the qualifier
carried by the fact and its evidence: the anatomical side ('in his
right hand'), the attachment or nearest relation ('on the price tag
closest to the kettle'), or the part location ('on the rear end of the
large blue trailer'). A bare 'in his hand', 'on the price tag', 'on the
sign', or 'on the vehicle' is forbidden in that situation. If the fact
and evidence carry no such qualifier while the notes show counterparts
exist, SKIP that fact.
- Never mention the crop viewport, cropping, truncation, cut-off or
out-of-view states, visibility limitations, or the VISIBILITY AND
REFERENCE NOTES section in any question; the notes only tell you when
an extra pinning qualifier is required. Positional image-frame
references such as 'near the left edge of the image' are locators, not
viewport talk, and remain allowed under the frame contract.
- FRAME CLAIMS COME FROM CAPTION1 ONLY. caption2 saw a crop, not the
photograph, so any left/right/side wording in it describes the crop's
own frame and is worthless | often actively wrong | as a statement
about the image: a crop taken from the right of the photo has its own
left side. Never copy or paraphrase a horizontal position out of
caption2 into a question. Use one only when caption1, which saw the
whole photograph, places the target there, and word it as caption1
does.
- NO FRAME SUPERLATIVES. Never write leftmost, rightmost, topmost,
bottommost, far left, far right, 最左, 最右, 最上, 最下, 最顶, 最底 about
the image in either language. Being the outermost of a category is a
claim about everything in the photograph, which neither caption can
establish; use an attachment, nearest-object, or part qualifier
instead. A superlative INSIDE one named host stays fine ('the bottom
word on the blue sign'), because the host bounds it.
- After eligible OCR facts, prefer counting / attribute facts, and
prefer facts whose relation_to_caption1 is "contradicted" or "unclear".
Output strict JSON only:
{"qas": [{"question_zh": "...", "question_en": "...",
"crop_question_zh": "...", "crop_question_en": "...", "answer": "...",
"question_type": "counting|ocr|attribute|action|spatial|comparison",
"evidence_sentence": "verbatim evidence", "source_fact_index": 0}]}
If no fact qualifies, output {"qas": []}.
```

LOCAL_ANCHOR_RULE_UNMARKED.

```
No artificial selection rectangle is present. Neither LOCAL question
may mention a box, outline, marker, or annotation.
```

LOCAL_ANCHOR_RULE_MARKED.

```
A thin red selection rectangle identifies the anchor in the crop. Both
LOCAL questions MUST explicitly use the rectangle only as a locator,
with wording such as 'the [object] inside the red selection rectangle'
/ '红框标出的[物体]'. If the fact concerns a part, content, marking, or
nearby object, relate it explicitly to that red-boxed anchor. Never ask
about the rectangle itself. GLOBAL questions must describe the
photographed anchor naturally and must not mention it.
```

J.5 JOINT CROP/FULL-IMAGE AUDIT AND REWRITING

EDGE_REWRITER.

```
You are the crop-boundary auditor and question rewriter of a visual-QA
data pipeline.

BACKGROUND | WHY THIS STAGE EXISTS:
The pipeline writes fine-grained QA about one anchor object. Every
earlier stage saw ONLY a crop taken around that anchor: a describer
captioned the crop, and a question writer turned caption facts into a
crop-local question plus a full-image question. Neither could see past
the crop's four edges, so two recurring failures escape them:
1. INCOMPLETE ANSWER CLAIM: the object or inscription carrying the
answer continues beyond the crop boundary, and the recorded answer
silently claims to be the complete text or a whole-object attribute
when it is not (a sign whose top line sits above the crop; a hull that
runs past the edge with more markings).
2. REFERENT AMBIGUOUS IN THE FULL IMAGE: the referring phrase picks
exactly one thing inside the crop but matches several things in the
full photograph (three white signs on the same wall; a second red
inscription on the same hull).
You are the first stage that sees BOTH views, so these become direct
observations for you. Downstream, independent solver models will answer
the crop-local question seeing only the crop, and the full-image
question seeing only the full photograph. Your goal: an honest, careful
solver in EITHER view must arrive at exactly the recorded answer.
Rejecting a question throws away everything already spent creating it,
so PREFER RESCUING a question with a bounded rewrite over dropping it |
but never rescue one at the cost of truthfulness.

INPUTS:
Image 1 is the crop. A cyan box labeled C1, if present, marks the
already-certified unique referent of the crop-local question; a thin
red rectangle, if present, marks the anchor. Both are annotations,
never scene content.
Image 2 is the FULL photograph with a red rectangle marking where the
crop was taken. Use it to see what continues beyond the crop and what
else in the scene matches the question's wording. Image 2 may be
lower-resolution: use it to detect continuation and count referent
matches, not to read new text.

THE QA UNDER AUDIT:
anchor: "{anchor}"
question_type: {question_type}
global question (full-image solvers) EN: {question_en}
global question ZH: {question_zh}
crop-local question EN: {crop_question_en}
crop-local question ZH: {crop_question_zh}
recorded answer: {answer}
```

```
mechanical OCR contract (null for non-OCR): {ocr_constraint}
caption evidence sentence: {evidence}

AUDIT PROCEDURE:
A. CONTINUATION: in Image 2, does the object or inscription that
carries the answer visibly continue beyond the crop rectangle in a way
that could add to or change the answer as the question words it? For an
OCR question, also check inside the crop whether the inscription's own
host is cut or occluded near the text so the text may continue out of
sight. A cut-off host is harmless when the question only asks about a
bounded part that is fully visible.
  THE BURDEN IS ON 'false'. Reaching the crop border IS continuation
  unless you can point to the reason it is not. Answer false only when
  you can name what bounds the host on that side | its own frame, edge,
  corner or end, visible INSIDE one of the two views | or when Image 2
  shows the whole host with nothing further. If the host runs to the
  border and you cannot see where it stops, that is true, not false,
  however legible the part you can read: a street plate whose left edge
  is the crop's left edge gets read as 'STATION ROAD' when the plate
  says 'OLD STATION ROAD', and a menu board whose bottom is the crop's
  bottom shows two of its five lines. Both look complete. Being able to
  read what is visible is not evidence that nothing else is there.
  Answering true does not condemn the QA | narrow_semantic rescues it
  by re-scoping to a sub-block you CAN see whole ('the line nearest the
  arrow marker', 'the bottom word'). Prefer that rescue over a confirm
  you cannot justify.
B. UNIQUENESS: read each referring phrase of the GLOBAL question and
count what it matches in Image 2; read the CROP question and count
matches in Image 1. A locator difference that a careful solver would
treat as decisive (unique attachment, position, or size wording) counts
as disambiguation; a bare category plus color usually does not.
   For an OCR question, ALSO count the text blocks: every separate
   inscription on the same host that a solver could take to match how
   the question describes the text (its color, case, or style) is a
   competing candidate. Judge like a strict solver, not a brand reader:
   a smaller second line counts; a darker shade of the same hue counts;
   digits and mixed letter-digit strings count as capital-glyph text; a
   registration or phone number printed under a brand name is the
   classic competing block. If more than one block could match, an
   honest solver cannot know which to read: the wording must single one
   out by row, size, or position (rewrite), or the QA must be dropped.
   Never assume the most prominent text is 'the' text.
   WEAK TEXT REFERENT RULE: a text referent described only by color
   and/or letter case ('the red capital letters') with no size, row, or
   position qualifier is weak wording. Whenever the host carries ANY
   other marking | even one you judge too small or too faint to read |
   prefer a minimal tightening rewrite (add 'the largest', a row, or a
   position word) over keep. The tightening costs nothing and makes the
   question robust to markings at the edge of legibility.
   Enumerate this audit in the output field other_inscriptions_on_host:
   one entry per OTHER inscription, print, number, or marking on the
   same host (empty list only when the host truly carries no other
   markings). could_match_question_wording=true means a strict solver
   could take that block as matching the question's text description.
   Keeping the question while any entry is true is a contradiction |
   then you must rewrite (single the target out by row, size, or
   position) or drop; code enforces this.
C. COUNT SCOPE (counting questions): the counted set must be closed by
visible boundaries; if Image 2 shows the set continuing outside the
crop while the question implies a total, the claim is broken.

VERDICTS:
- keep: both audits clean; the recorded answer is exactly what an
honest solver would give in both views.
```

```
- rewrite: a bounded edit repairs the problem | tighten a referring
phrase so it is unique in both views, or narrow an OCR claim to a safe
span (see OCR SPAN DECISION). Rewrite all four question fields
consistently.
- drop: the recorded answer is factually wrong or unverifiable in
either view, no crop-visible qualifier can make the referent unique, or
a count's scope is opened by the crop edge. Also drop when a rescue
would need information a crop-only solver cannot see.
- COUNT RESCUE BAN: a counting question whose set the crop opens can
NEVER be rescued by wording. Adding 'visible' / '可见' or similar makes
the crop total and the full-image total differ, so the two views would
have different correct answers. Such a question is unfixable: drop it.

REWRITE RULES:
- The recorded answer may NEVER be extended, corrected, or replaced
with content you read in Image 2 | you sit downstream of evidence
extraction and may only NARROW claims. For a non-OCR question the
answer must stay exactly unchanged; you edit wording only. For an OCR
question the answer may only shrink to a contiguous part of the
recorded sequence.
- CROSS-VIEW LOCATOR ATTRIBUTES: check every attribute the question
uses to FIND the target | color, size, material, shape | against both
images, including attributes the draft already contained. A sliver
visible in the crop routinely misdescribes the whole object in the full
photograph: a white sign whose only visible strip sits against an
orange panel reads as 'the orange sign' in the crop and matches nothing
(or something else) in the full image. Replace a misleading attribute
with one true in both views, or drop when none exists.
- Every qualifier you add must be verifiable inside the CROP (crop
solvers cannot see past it) and stay true in the full image. Prefer
attachment, nearest-object, part-position, row/line position, or
relative-size cues. Never use viewport words (crop, edge, truncated,
cut off, visible portion, 画面边缘, 被截断), and never copy answer
characters or answer words into any question field.
- Keep the two languages synchronized and keep the global/crop pairing:
the global question locates the target for a full-image solver (usually
via the anchor); the crop question drops context the crop cannot
establish.
- Spatial wording in rewrites must obey this contract:
<<<INSERT: SPATIAL_FRAME_RULES>>><<<INSERT: CROP_FRAME_CAVEAT>>>
OCR SPAN DECISION (REQUIRED for every ocr question, in
ocr_span_decision; null for non-OCR):
You MAY change an OCR question's answer | this is usually the BEST
rescue, not a last resort | provided the new answer is a contiguous
part of the recorded one. Upstream drafts are often clumsy (a
20-character positional span, or a sequence that silently spans a
truncation); your job is to re-scope onto a piece that is easy to refer
to and true in both views: the bottom word, the largest line, the top
row, or the first few letters. Work through the decisions in this order
and pick the FIRST that yields a claim true in both views:
- confirm: the inscription the question asks about is completely
visible with no plausible continuation, and the recorded answer is its
complete text (or the question already carries consistent span
wording). Valid with verdict keep or with a rewrite that only fixed the
referring phrase. confirm is FORBIDDEN whenever you observed the
inscription's host cut or continuing at or near the text region | in
the crop or in Image 2. Observing continuation and confirming anyway is
a contradiction; choose narrow_semantic, demote, or drop instead.
```

```
- narrow_semantic (PREFERRED rescue): re-scope the question to one
fully-visible, visually isolable sub-block | the bottom word, the
topmost row, the largest line, the group on one panel | whose complete
text is a contiguous whole-word part of the recorded sequence; the
answer becomes exactly that part ('what is the bottom word on the sign'
-> 'Bakery' where the recorded answer was 'Old Town Bakery'). A solver
reads such a unit naturally, with no character counting. Identify the
sub-block ONLY by visual properties (position, size, row, color of its
panel), never by quoting its text.
  THE SUB-BLOCK MUST BE CLOSED ON THE CUT SIDE. A sub-block is only
  'fully visible' if every one of its own boundaries is inside the
  crop, and the boundary that matters is the one facing the cut. When
  the host is cut on its LEFT or RIGHT, a row or line runs straight
  into that cut and is not closed, so 'the lower line', 'the word at
  the top of the sign', 'the largest word on the second row' and any
  other row- or band-based selector is invalid there | pick a unit
  bounded away from the cut instead (a word whose far side is clear
  space, the block on one whole panel). When the host is cut at the TOP
  or BOTTOM, rows away from the cut are fine and a column is not. A
  size superlative ('the largest word') over a unit that continues out
  of view is a claim about text you cannot see. If no closed sub-block
  exists, use demote_first / demote_last or drop; do not narrow onto an
  open one.
- demote_first / demote_last: for the cases narrow_semantic cannot
express | no whole word is fully safe, but a clean sub-word boundary
fragment is (only 'SUR' of a longer word is clearly visible). Ask for
exactly the first/last k alphanumeric characters of the recorded
sequence, with 3 <= k <= 7 | solvers must COUNT the characters, so a
larger k turns reading into error-prone counting and is mechanically
rejected. Use demote_first when the text's beginning boundary is
visibly real and continuation risk is at the end; demote_last in the
mirror case. CROSS-VIEW SPAN AGREEMENT (mandatory): before choosing k,
read the same characters in Image 2 | every one of the k characters you
keep must be identical in the crop and in the full photograph. A
recorded sequence stitched together across a truncation (two cut lines
read as one string) diverges from the real text at the stitch, and any
k that reaches the stitch makes the answer wrong for full-image
solvers: choose the smallest stable k (>= 3) that stays strictly before
the divergence, and if even 3 agreeing characters do not exist, demote
is unavailable. Both English question fields must contain the explicit
k right beside first/last (e.g. 'the first 3 characters'), and the
Chinese fields must mirror it (e.g. \前3个字母数字字符"). The answer is
that exact span with its original spacing and punctuation preserved.
- SHORT ANSWERS: an answer of fewer than 3 alphanumeric characters
cannot be narrowed any further, so it survives only as confirm with no
continuation anywhere at that text. If the inscription continues out of
view (or its host is cut at the text), a 1-2 character recorded answer
is a fragment of something longer: drop it rather than re-scoping the
question onto that fragment | code rejects such a rescue either way.
- Drop an OCR question only AFTER checking that no narrowing exists:
neither a whole-word sub-block (narrow_semantic) nor a 3-7 character
boundary span (demote) yields a claim true in both views. A recorded
answer being wrong AS A WHOLE does not justify drop when a narrowed
part of it is still exactly right.

Do not transcribe scene text in reasons beyond what the recorded answer
already contains. Return strict JSON only:
```

```
{"continuation_beyond_crop": true|false, "full_image_referent_matches":
<integer count for the global question's referent in Image 2>,
"crop_referent_matches": <integer count in Image 1>,
"other_inscriptions_on_host": [{"description": "where/what kind of
marking, WITHOUT transcribing it", "could_match_question_wording":
true|false}] (for non-OCR questions use []), "verdict":
"keep|rewrite|drop", "ocr_span_decision":
"confirm|narrow_semantic|demote_first|demote_last"|null, "rewrite":
null|{"question_zh": "...", "question_en": "...", "crop_question_zh":
"...", "crop_question_en": "...", "answer": "..."}, "drop_reason":
null|"concise reason", "reason": "concise image-specific audit reason"}
```

J.6 CROP, FULL-IMAGE, AND BLIND ANSWERING

CROP_VQA.

```
Look at this image and answer the question with a single short value (a
number, a color, a text string, or a few words). No explanation.
Question: {question}
```

CROP_VQA_ANCHOR_MARKED.

```
The thin red selection rectangle is an artificial locator identifying
the photographed anchor selected by the pipeline. A crop-local question
may refer to the object inside the red rectangle, its parts, contents,
markings, or directly related nearby objects. Use the rectangle only to
resolve that reference rather than unmarked same-category objects.
Never treat the rectangle itself as photographed content or as the
requested answer.
<<<INSERT: CROP_VQA>>>
```

CROP_OCR_VQA.

```
Look at this image and answer the OCR question by transcribing only the
requested text or character span. Preserve character order and do not
infer characters that are not clearly visible. If the requested text is
not clear, answer exactly: unknown. If the question asks for exactly N
characters or letters from a specified position, return exactly N
alphanumeric characters in normal reading order: mechanically take the
final N for a last/rightmost/ending-N question or the first N for a
first/leftmost/beginning-N question. Spaces and punctuation do not
count toward N. Do not return the complete visible word or any extra
character. Check that the response contains exactly N characters before
answering. Return the answer only, with no explanation.
Question: {question}
```

CROP_OCR_VQA_ANCHOR_MARKED.

```
The thin red selection rectangle is an artificial locator identifying
the photographed anchor selected by the pipeline. A crop-local question
may refer to the object inside the red rectangle, its parts, contents,
markings, or directly related nearby objects. Use the rectangle only to
resolve that reference rather than unmarked same-category objects.
Never transcribe or return the rectangle itself as OCR content.
<<<INSERT: CROP_OCR_VQA>>>
```

FULL_IMAGE_VQA.

```
Look at this image and answer the question with a single short value (a
number, a color, a text string, or a few words). If you cannot see the
answer clearly, answer exactly: unknown. No explanation.
Question: {question}
```

FULL_IMAGE_OCR_VQA.

```
Look at this image and answer the OCR question by transcribing only the
requested text or character span. Preserve character order and return
your single best transcription from the visible image, even when some
characters are difficult to read. If the question asks for exactly N
characters or letters from a specified position, return exactly N
alphanumeric characters in normal reading order: mechanically take the
final N for a last/rightmost/ending-N question or the first N for a
first/leftmost/beginning-N question. Spaces and punctuation do not
count toward N. Do not return the complete visible word or any extra
character. Check that the response contains exactly N characters before
answering. Return the answer only, with no explanation.
Question: {question}
```

BLIND_ANSWER.

```
You cannot see the photograph. Answer the question using only its
wording and your learned language and world knowledge. Even if the
image-dependent answer is uncertain, you must make your single most
likely guess. Return one concrete, short answer of the requested type.
Never answer unknown, cannot determine, not enough information, or any
other abstention. Do not provide alternatives or an explanation.
Question: {question}
```

ANSWER_EQUIVALENCE.

```
You are a careful, strict, and deterministic evaluator of short
visual-question answers.

Inputs:
- Question: {question}
- Reference answer: {reference}
- Candidate answer: {candidate}

Decide whether the candidate correctly fills the exact answer slot
requested by the question.

Internally follow this order:
1. Infer the requested answer type and every explicit constraint in the
question.
2. Identify the single answer asserted by the candidate. If it gives
incompatible alternatives or adds an extra incorrect answer, reject it.
3. Compare that asserted answer with the reference in meaning.

Accept harmless grammatical context, articles, synonyms, paraphrases,
and a more specific description when it preserves the reference
meaning. Do not require the same wording. Reject a merely related
object or attribute, a contradiction, the wrong answer type, an
incomplete answer, or extra incorrect factual content.
For ordinary VQA, judge semantic equivalence in the context of the
question; extra location or object context is harmless when it does not
change the requested answer.

Think through the comparison internally. Output exactly one tag and
nothing else:
<verdict>yes</verdict>
or
<verdict>no</verdict>
```

OCR_ANSWER_EQUIVALENCE.

```
You are judging whether a candidate OCR response correctly answers the
exact question. Judge visual content, not answer-only formatting.
Question: {question}
Reference answer: {reference}
Candidate response: {candidate}
Apply every positional and length constraint in the question. In
particular, if it asks for the first/leftmost/beginning or
last/rightmost/ending N characters and the candidate gives a longer
transcription, mechanically take the first or final N alphanumeric
characters from that transcription before comparing it with the
reference. An explanation is allowed when its transcription and stated
answer are mutually consistent. Do not say yes merely because the
reference text appears somewhere in an unrelated explanation. If the
response gives conflicting transcriptions or answers, or the derived
requested span differs by any extra, missing, reordered, substituted,
or accented character, say no. Ignore only letter case and superficial
spacing or punctuation. Reply with exactly one word: yes or no.
```

J.7 MULTI-SOURCE DISTRACTORS AND MCQ VERIFICATION

FULL_DISTRACTOR_CLUSTERING.

```
You are a conservative semantic auditor constructing multiple-choice
distractor candidates from repeated full-image VQA responses. You do
not see the image and must not decide which answer is visually true. Do
all semantic normalization and clustering yourself; the caller only
validates IDs and counts support mechanically.

The caller has already removed responses judged equivalent to the
reference and responses that are literally just 'unknown'. Some
remaining responses may still be verbose abstentions, conflicted
answers, or judge errors. Audit them conservatively rather than
assuming every input is a usable distractor.

First infer the exact answer slot requested by the question: for
example count, OCR string, time, color, material, object identity,
action, or spatial relation. Then audit every response independently.

Response rules:
1. committed means the response ultimately selects exactly one concrete
answer. Extract the final selected answer even if it follows reasoning
or an explicit self-correction.
2. abstain means the response ultimately says unknown, unclear,
unreadable, cannot determine, or otherwise declines to answer. Classify
it as abstain even if it mentions speculative possibilities.
3. conflicted means it leaves multiple incompatible alternatives
without selecting one. Do not arbitrarily retain one alternative.
4. For abstain or conflicted, extracted_answer and cluster_id must be
null.

Clustering rules:
5. Cluster committed answers only when they are interchangeable answers
to this exact question. Normalize spelling, morphology, number words,
abbreviations, and harmless surrounding wording semantically.
6. Do not merge merely related answers. Different counts, actions,
times, directions, materials, object identities, or OCR strings are
different answers.
7. relation_to_reference is equivalent only when the cluster would
answer this question with the same meaning as the reference. This check
remains required because the earlier per-response judge can make
mistakes.
```

```
8. answer_slot_fit is valid only when the cluster supplies the
requested kind of answer. A color is invalid for a material question;
an explanation is acceptable only when one clean final answer can be
extracted.
9. canonical_answer must be a concise normalized form of the cluster's
shared meaning. representative_id must name one actual member whose
answer supports that canonical form.

Completeness contract:
- Process every supplied response ID exactly once.
- Every committed response belongs to exactly one cluster.
- Cluster member_ids must exactly partition all committed responses.
- Never invent, omit, or duplicate an ID.
- Output one strict JSON object and no Markdown.

Question type: {question_type}
Question: {question}
Reference answer: {reference}

Responses:
{responses}

Output schema:
{"answer_slot": "short description of what kind of answer is
requested", "responses": [{"id": "F001", "disposition":
"committed|abstain|conflicted", "extracted_answer": "answer or null",
"cluster_id": "C1 or null"}], "clusters": [{"cluster_id": "C1",
"canonical_answer": "concise answer", "member_ids": ["F001"],
"representative_id": "F001", "relation_to_reference":
"equivalent|different", "answer_slot_fit": "valid|invalid", "reason":
"short explanation"}]}
```

BLIND_DISTRACTOR_CLUSTERING.

```
You are a conservative semantic auditor constructing multiple-choice
distractor candidates from blind, text-only model responses. You do not
see the image. Do all semantic normalization and clustering yourself;
the caller will only validate IDs and count support mechanically.

First infer the exact answer slot requested by the question: for
example count, OCR string, time, color, material, object identity,
action, or spatial relation. Then audit every response independently.

Response rules:
1. committed: the response ultimately selects exactly one identifiable
answer. Extract that final answer even if it is surrounded by
reasoning.
2. abstain: it says unknown, cannot determine, or refuses to guess.
3. conflicted: it leaves multiple incompatible alternatives without
selecting one. A phrase such as 'red or blue' is conflicted when the
question asks for one color. Do not arbitrarily keep one option.
4. For abstain or conflicted, extracted_answer and cluster_id must be
null.

Clustering rules:
5. Cluster committed answers only when they are interchangeable answers
to this exact question. Normalize spelling, morphology, number words,
abbreviations, and harmless extra wording semantically.
6. Do not merge merely related answers. Running, sitting, reading, and
waving are different actions unless the response itself clearly commits
to the same one. ALPHA, BETA, and GAMMA are different OCR strings.
These are abstract boundary examples only; do not copy them into the
result.
7. relation_to_reference is equivalent only when the cluster would
answer the question with the same meaning as the reference.
```

```
8. answer_slot_fit is valid only when the cluster supplies the
requested kind of answer. A color is invalid for a material question;
an explanation is acceptable only when one clean final answer can be
extracted.
9. canonical_answer must be a concise normalized form of the cluster's
shared meaning. representative_id must name one actual member whose
answer supports that canonical form.

Completeness contract:
- Process every response ID exactly once.
- Every committed response belongs to exactly one cluster.
- Cluster member_ids must exactly partition all committed responses.
- Never invent, omit, or duplicate an ID.
- Output one strict JSON object and no Markdown.

Question type: {question_type}
Question: {question}
Reference answer: {reference}

Responses:
{responses}

Output schema:
{"answer_slot": "short description of what kind of answer is
requested", "responses": [{"id": "B001", "disposition":
"committed|abstain|conflicted", "extracted_answer": "answer or null",
"cluster_id": "C1 or null"}], "clusters": [{"cluster_id": "C1",
"canonical_answer": "concise answer", "member_ids": ["B001"],
"representative_id": "B001", "relation_to_reference":
"equivalent|different", "answer_slot_fit": "valid|invalid", "reason":
"short explanation"}]}
```

VLM_DISTRACTOR_GENERATOR.

```
You are a conservative visual multiple-choice distractor generator. You
see exactly one {view_name} image. Generate zero to three short
candidate answers that are visually motivated yet definitely incorrect
for the question's intended target. The reference answer is immutable
and correct. Returning no candidate is better than inventing
unsupported content.

Infer the exact answer slot and the semantic core of the reference
first. Every candidate must fill the same slot naturally and must
differ in that core; changing an incidental adjective does not create a
different answer. For object identity, use a genuinely different head
noun, not another colored instance of the reference object. For
actions, use a concrete visible action rather than clothing,
possession, appearance, or posture. For counts, use only a defensible
alternate inclusion boundary or nearby same-class group. For
spatial/person identity, preserve the requested relation or entity
type.

Preferred visual strategies, strongest first:
1. same_property_other_instance: the requested property on another
similar visible object.
2. analogous_relation_other_object: a genuinely different object
visible in an analogous relation or location.
3. target_other_attribute: only for scalar attributes such as color,
material, or pattern, use another clearly visible value on the target.
4. nearby_same_slot: a clearly visible value of the exact requested
type from a nearby object or region.
5. count_near_miss: another visually defensible count, never an
arbitrary number.
```

```
A different visible source is positive evidence: for a blue-hat color
question, a yellow shirt or another person's white hat may supply
yellow or white. Name the exact source and relation. Do not use world
knowledge, generic alternatives, likely/perhaps/resembling evidence,
unknown, the reference, or a synonym of it. MUTUAL EXCLUSIVITY: a
candidate must be necessarily false whenever the reference is true, and
an average viewer must be able to refute it by looking at the image.
Never propose a broader or narrower rewording of the reference (stripes
when the reference is white stripes, metal when it is stainless steel),
a logically nested category (rectangular when the reference is square |
every square is a rectangle), or a value so perceptually close to the
reference that the image cannot distinguish them (square vs rectangle,
circle vs near-circular oval). The thin red rectangle, if present, is
an artificial locator and never image content. Rank strongest first, do
not pad to three, and output strict JSON only.

View discipline:
{view_rules}

Question type: {question_type}
Question: {question}
Immutable reference answer: {reference}

Output schema:
{"candidates": [{"candidate_id": "V1", "answer": "short answer",
"strategy": "strategy name", "evidence_description": "visible fact and
location", "source_object": "source object or region",
"relation_to_target": "why it belongs elsewhere", "quality":
"high|medium|low"}], "no_candidate_reason": "reason or null"}
```

VISUAL_DISTRACTOR_CROP_RULES.

```
Use only information visible in this expanded crop. Prefer a
same-property value from another local instance, then another same-type
attribute. Do not assume content from the unseen full image.
```

VISUAL_DISTRACTOR_FULL_RULES.

```
Use only information visible in the native full image. Borrow from a
different object or text region and match the requested answer slot as
closely as possible. You do not have access to the zoom crop.
```

VLM_DISTRACTOR_VALIDATOR.

```
You are an independent conservative visual validator for proposed
multiple-choice distractors. You see the same {view_name} image used by
the generator. The immutable reference is correct. Evaluate every
candidate independently and do not generate replacements. A candidate
should be wrong for the intended target but may deliberately borrow a
value from a different visible object, attribute, action, count
boundary, person, or text region. Do not reject merely because the
source is elsewhere.

Infer the exact answer slot and reference semantic core. For scalar
color, material, pattern, or orientation, a directly visible value from
any source object is allowed. For object identity, reject any candidate
sharing the reference head noun/category regardless of color, owner, or
instance; accept only a genuinely different object in an analogous
relation. For action, accept a concrete action visibly performed
elsewhere but reject clothing, appearance, wearing, possession, and
static attributes. For counts, require a defensible visible
group/boundary. For spatial/person identity, preserve the requested
relation/entity type.
```

```
For OCR, independently transcribe the claimed source instead of
trusting the evidence description. The candidate must be an exact
complete token or a question-licensed first/last-N span of that
transcription; the full source may be longer, so PORTUGAL may be
supported by LICOR DE PORTUGAL. A one-word question requires one word,
and a name question rejects a bare route/count number, but another
visible word need not share the target's real-world meaning. Source
color and capitalization affect quality, not whether the answer string
itself fits. If the text is too small to read independently, use reject
or uncertain.

Accept only when the candidate has the same answer type/granularity,
differs from the reference core, is not also correct for the target,
has verified visual evidence, uses a valid strategy, and is
concise/non-unknown. MUTUAL EXCLUSIVITY IS MANDATORY: 'not equivalent
to the reference' is NOT enough. Reject any candidate that could be
true together with the reference | broader/narrower rewordings (stripes
vs white stripes), logically nested categories (rectangular when the
reference is square), or values the image cannot refute because they
are perceptually adjacent to the reference (square vs rectangle, circle
vs near-circular oval; classic color pairs: silver vs white or grey,
gold vs yellow, beige or cream vs white). The test: could a careful
viewer point at the image and say 'definitely not this'? If not,
reject. Reject generic, inferred, likely, or world-knowledge evidence.
decision=accept requires quality high or medium. Process every ID
exactly once and output strict JSON only.

View: {view_name}
Question type: {question_type}
Question: {question}
Immutable reference answer: {reference}

Candidates:
{candidates}

Output schema:
{"answer_slot": "exact requested slot", "reference_semantic_core":
"core meaning", "evaluations": [{"candidate_id": "V1", "decision":
"accept|reject|uncertain", "slot_fit": true,
"different_from_reference": true, "not_correct_for_target":
"yes|no|unclear", "visual_source_verified":
"yes|no|unclear|not_applicable", "strategy_valid": true,
"ocr_length_constraint_satisfied": "yes|no|not_applicable",
"independent_source_transcription": "exact text or null",
"source_matches_candidate": "yes|no|unclear|not_applicable", "quality":
"high|medium|low", "reason": "specific concise reason"}]}
```

DISTRACTOR_ONLY_REWRITE.

```
You are editing multiple-choice distractor options. Rewrite ONLY the
listed distractors. The reference answer is immutable context: never
rewrite it, replace it, include it in the output list, or turn any
distractor into the reference answer. You do not see the image.

Question type: {question_type}
Question: {question}
Immutable reference answer: {reference}

Distractors to edit:
{distractors}

For each distractor:
1. Preserve exactly the answer it commits to. The rewrite must remain
semantically interchangeable with that distractor for this question.
2. Make it a concise, natural, standalone answer to the question.
Remove reasoning, hedging, redundant framing, and sentence-level
filler.
3. Match the grammatical form and surface style of the immutable
reference when useful, without changing the reference itself. For
example, if the reference spells a count as a number word, spell
distractor counts as number words too.
4. Never change a count value, time, OCR character sequence, proper
name, direction, color, material, negation, object identity, action, or
spatial relation. Formatting-only normalization is allowed when meaning
is unchanged.
5. Do not add visual details, combine options, explain why an option is
wrong, or mention the image or reference answer.
6. If a distractor is already concise, natural, and style-compatible,
copy it unchanged.
7. Process every distractor ID exactly once. Do not add, drop, merge,
or reorder IDs. Output strict JSON only.

Output schema:
{"distractors": [{"id": "D001", "rewritten_answer": "concise answer",
"changed": true, "reason": "brief editing reason"}]}
```

MULTIPLE_CHOICE_POOL_CURATOR.

```
You are the final text-only curator for a visual multiple-choice
question. The immutable reference answer is correct. The entries come
from several already-audited sources: jointly validated crop/full
visual distractors, real wrong full-image model answers, qualified
blind-prior answers, and possibly Qwen-only visual reserve candidates.
You do not see the image. Do semantic organization only; do not invent,
rewrite, improve, split, or add answer strings.
```

```
Treat provenance as evidence that an answer is a serious candidate,
never as a reason to keep two semantically identical options. A
borrowed scalar value remains allowed when it fits the requested slot:
for a hat-color question, yellow may come from a visible yellow shirt.
Infer the exact answer slot from the question and reference. Process
every entry ID exactly once. Mark an entry eligible only when its
answer string itself is a concise, committed answer in that exact slot
and semantic granularity, semantically different from the reference,
and not an abstention, explanation, or malformed fragment. Reject a
related but different component, attribute, or category: for example,
black rubber tires do not fit a wheel-style slot whose reference is
silver alloy wheels, while spoked wheels do. Image correctness was
checked upstream; this step judges only answer-slot fit, reference
equivalence, mutual exclusivity with the reference, and cross-entry
equivalence. A surviving distractor must be NECESSARILY FALSE when the
reference is true: route to reference_equivalent any broader, narrower,
or logically nested description that can be true together with the
reference (stripes when the reference is white stripes; rectangular
when the reference is square | every square is a rectangle; metal when
the reference is stainless steel), and route to invalid any entry an
average viewer could not definitively refute because it is perceptually
adjacent to the reference (square vs rectangle, circle vs near-circular
oval). Use reference_equivalent for paraphrases, aliases,
broader/narrower wording that would still be accepted as the correct
answer, numeric/number-word equivalents, and answers that merely add an
attribute to the same requested object identity. Use invalid for wrong
answer type, unknown/uncertain text, or content that does not stand
alone as an option.

Cluster every eligible entry by the answer it commits to for this exact
question. Merge semantic equivalents across all sources even when
wording, articles, capitalization, inflection, number format, or
specificity differ. Do not merge merely related alternatives that would
be distinct choices. Each eligible ID must occur in exactly one
cluster; rejected IDs occur in none. Choose representative_id from the
cluster itself, favoring the most concise natural existing answer.
Never output a new canonical answer.

Question type: {question_type}
Question: {question}
Immutable reference answer: {reference}

Pool entries:
{entries}

Output strict JSON only:
{"answer_slot": "exact requested slot", "entries": [{"id": "P001",
"disposition": "eligible|reference_equivalent|invalid", "cluster_id":
"C1 or null", "reason": "concise semantic reason"}], "clusters":
[{"cluster_id": "C1", "member_ids": ["P001"], "representative_id":
"P001", "relation_to_reference": "different", "answer_slot_fit":
"valid", "reason": "shared committed answer"}]}
```

MULTIPLE_CHOICE_SOLVER.

```
Answer one multiple-choice visual question from the supplied zoom crop.
The global question gives full-scene identity context; the crop
question asks about the same target using wording appropriate for this
crop. Use the crop question as the direct visual task. Inspect the
image carefully and select exactly one displayed option. Do not
explain, hedge, list multiple options, or use outside knowledge when
the crop provides the answer.

Question type: {question_type}
Global question: {global_question}
Crop question: {crop_question}
Options:
{options}

Return strict JSON only, using one supplied choice label:
{"choice": "A"}
```