ORIC: Benchmarking Object Recognition under Contextual Incongruity in Large Vision-Language Models

Supplementary Material

A. ORIC Method, Analysis, and ORIC-Bench Evaluation Metrics

A.1. LLM-Guided Sampling Method (Positive Question Construction)

Algorithm 1 Positive Question Construction

Require: Image I, objects $\mathcal{O} = \{(n_i, B_{ij})\}$, integer k
Ensure: Positive question Q
1: **for** $i = 1$ to N **do**
2: $A_i \leftarrow \text{area}(\bigcup_j B_{ij})$
3: **end for**
4: Sort $\mathcal{O}$ by A_i (descend.)
5: $\mathcal{O}_{\text{ROI}} \leftarrow$ bottom 50%, $\mathcal{O}_{\text{nonROI}} \leftarrow$ top 50% ▷ Note: Objects exactly at the 50% boundary are classified as non ROI.
6: $\mathcal{C} \leftarrow \emptyset$
7: **for** $o \in \mathcal{O}_{\text{ROI}}$ **do**
8: **if** LLM says "no" for o given $\mathcal{O}_{\text{nonROI}}$ **then**
9: $\mathcal{C} \leftarrow \mathcal{C} \cup \{o\}$
10: **end if**
11: **end for**
12: Randomly pick k objects from $\mathcal{C}$ as Q **return** Q

Fig. 6 presents the prompt used in LLM-guided rejection sampling for constructing positive questions in the ORIC. Specifically, {`background_objects`} serves as a placeholder for all non-ROI objects. For example, if there are three non-ROI objects, they could be represented as `["car", "person", "bottle"]`. Meanwhile, {`target_object`} represents a placeholder for a specific ROI object, such as `"vase"`.

LLM-Guided Rejection Sampling

Given the following background objects: {`background_objects`}, can you determine whether the following target object {`target_object`} is present in the image with relying on textual priors, common-sense knowledge, or general assumptions about object co-occurrences?
Please respond with yes or no.

Figure 6. **Prompt for LLM-guided rejection sampling.** {`background_objects`} is a placeholder for all non-ROI objects, and {`target_object`} denotes a specific ROI object.

A.2. CLIP-Guided Sampling Method (Negative Question Construction)

Algorithm 2 Negative Question Construction

Require: Query image I_q, candidate images $\{I_1, \ldots, I_n\}$, non-existent objects $\mathcal{O}_{\text{non}} = \{n_i\}_{i=1}^{M}$, integer k
Ensure: Negative question Q
1: Select the most similar image:

$$I' = \arg\min_{I_i \in \mathcal{I}} \left(1 - \frac{\mathbf{e}_q \cdot \mathbf{e}_i}{\|\mathbf{e}_q\| \|\mathbf{e}_i\|}\right)$$

2: **for** $i = 1$ to M **do**
3: Construct text: $T_i \leftarrow$ "an image contains { n_i}"
4: Compute CLIP score: $s_i \leftarrow \text{CLIPScore}(I', T_i)$
5: **end for**
6: Sort $\{n_i\}$ by s_i (descending)
7: Select top k objects: $\mathcal{S} \leftarrow \{n_{i_1}, \ldots, n_{i_k}\}$
8: Construct Q using $\mathcal{S}$ **return** Q

A.3. Image Similarity Analysis via Minimum Distance

To further characterize the ORIC, we analyzed the visual relationships between positive and negative questions through image similarity measurements. Specifically, for each object class appearing in positive ("yes") questions, we computed its minimum visual distance to negative ("no") questions containing the same object class. Given an object o_i, let the set of positive images be $\mathcal{I}_i^+ = \{I_{i,1}^+, \ldots, I_{i,m}^+\}$ and the set of negative images be $\mathcal{I}_i^- = \{I_{i,1}^-, \ldots, I_{i,n}^-\}$. We extracted visual feature vectors using a ViT encoder and computed pairwise cosine distances as follows:

$$D(I_{i,k}^+, I_{i,l}^-) = 1 - \frac{e(I_{i,k}^+) \cdot e(I_{i,l}^-)}{\|e(I_{i,k}^+)\| \, \|e(I_{i,l}^-)\|} \tag{7}$$

where $e(\cdot) = \text{ViT}(\cdot)$ denotes the ViT feature extractor. The minimum distance between positive and negative sets is defined as $D_{\min} = \min_{k,l} D(I_{i,k}^+, I_{i,l}^-)$. To ensure thorough evaluation, we calculated these minimum distances using three widely used vision encoders commonly employed in encoder-based LVLMs: CLIP-ViT-BigG-P14, SigLIP-SO400M-P14-384 [84], and EVA02-CLIP-BigE-P14 [62]. These analyses highlight the distinctiveness of ORIC in capturing contextually challenging object recognition scenarios compared to existing benchmarks. In Tab. 6, questions generated from ORIC shows consistently smaller minimum cosine distances between "yes" and "no" samples than POPE

across all three vision encoders. This suggests greater visual similarity between positive and negative examples, making object recognition more challenging and realistic.

Vision Encoder	POPE	ORIC
CLIP-ViT-BigG-P14	0.37	**0.14**
SigLIP-SO400M-P14-384	0.28	**0.11**
EVA02-CLIP-BigE-P14	0.40	**0.13**

Table 6. **Comparison of Minimum Cosine Distances.** This table compares the minimum cosine distances between positive and negative questions across three vision encoders. A smaller distance indicates greater semantic similarity between images, meaning "yes" and "no" questions are linked to finer image details and higher representational clutter, making object recognition more challenging and realistic.

A.4. Evaluation Metric Formulas

For a binary classification problem with labels *yes* and *no*, we define the following terms:

- **TP** (True Positive): Number of samples correctly predicted as *yes* (Ground Truth: *yes*).
- **TN** (True Negative): Number of samples correctly predicted as *no* (Ground Truth: *no*).
- **FP** (False Positive): Number of samples incorrectly predicted as *yes* (Ground Truth: *no*).
- **FN** (False Negative): Number of samples incorrectly predicted as *no* (Ground Truth: *yes*).

The performance metrics include accuracy, the proportion of *yes* predictions, macro precision, recall, and F1 score. These are defined as follows:

Class-wise Metrics:

$$\text{Precision}_{\text{yes}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Recall}_{\text{yes}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$F1_{\text{yes}} = 2 \times \frac{\text{Precision}_{\text{yes}} \times \text{Recall}_{\text{yes}}}{\text{Precision}_{\text{yes}} + \text{Recall}_{\text{yes}}} \tag{10}$$

$$\text{Precision}_{\text{no}} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{11}$$

$$\text{Recall}_{\text{no}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{12}$$

$$F1_{\text{no}} = 2 \times \frac{\text{Precision}_{\text{no}} \times \text{Recall}_{\text{no}}}{\text{Precision}_{\text{no}} + \text{Recall}_{\text{no}}} \tag{13}$$

Macro-averaged Metrics:

$$\text{Precision}_{\text{macro}} = \frac{\text{Precision}_{\text{yes}} + \text{Precision}_{\text{no}}}{2} \tag{14}$$

$$\text{Recall}_{\text{macro}} = \frac{\text{Recall}_{\text{yes}} + \text{Recall}_{\text{no}}}{2} = \text{Accuracy} \tag{15}$$

Since our experimental datasets are all balanced, the number of positive and negative samples is equal. In this case, Accuracy $=$ $\text{Recall}_{\text{macro}}$ because accuracy measures the overall proportion of correctly classified samples, and macro recall, being the unweighted average of recall for both classes, reflects the same value.

$$F1_{\text{macro}} = \frac{F1_{\text{yes}} + F1_{\text{no}}}{2} \tag{16}$$

Proportion of Yes Predictions: The proportion of "yes" predictions (i.e., the percentage of all predictions that are classified as "yes") is given by:

$$\text{Yes Proportion} = \frac{\text{TP} + \text{FP}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{17}$$

B. ORIC-Bench Experiment and Analysis

B.1. Evaluated Models

We evaluate **18** widely used LVLMs spanning both encoder-based and encoder-free architectures. The encoder-based models include Qwen3-VL-8B-Instruct [3, 4], SmolVLM2-2.2B-Instruct [49], InternVL3-9B [88], Kimi-VL-A3B-Instruct [64], Janus-Pro-7B [9], Llama-3.2-11B-Vision [12], LLaVa-v1.6-7B [42], Phi-3.5-Vision-Instruct [1], Molmo-7B-D-0924 [16], GLM-4V-9B [22], Chameleon-7B [63], VILA-1.5-13B [39], and BLIP3 [74]. Encoder-free models include Fuyu-8B [5], EVE-7B-HD-v1.0 [17], Emu3-Chat [69], and the closed-source GPT-5 [60]. What's more, we benchmark against **2** open-vocabulary detection models: Grounding DINO 1.5 Pro [57] and OWLv2 [51].

B.2. Prompt Templates of Experiments

Large Vision-Language Models (LVLMs) Fig. 7 illustrates the prompt used for LVLMs in both the POPE and LOPE-3 benchmarks. An example of a specific question is: *"Is there a person in the image?"*.

LVLMs

`<image>`
Question: {`question`}
Please answer the question based on the given image.

Figure 7. **The Prompt of LVLMs.** The prompt of a binary classification task for LVLMs is used in all experiments, where {question} serves as a placeholder for a specific query and `<image>` is the placeholder for a specific image.

We use four distinct prompts in our experiments, detailed below:

- Is there {object} in the image?
- Does the image contain {object}?
- Have you noticed {object} in the image?
- Can you see {object} in the image?

The {object} is the placeholder for a detail object.

Grounding DINO 1.5 Pro Prompt: Fig. 8 shows the prompt for Grounding DINO 1.5 Pro. For example, if an image contains four unique objects—sports ball, person, car, and traffic light—the corresponding prompt would be: *"sports ball.person.car.traffic light"*.

Grounding DINO 1.5 Pro

$\{object1\}.\{object_2\}.\cdots.\{object_n\}$

Figure 8. **The Prompt of Grounding DINO 1.5 Pro.** The prompt used for the binary classification task in all experiments with Grounding DINO 1.5 Pro follows a dot-separated notation to specify multiple objects. Placeholders $\{object1\}, \{object_2\}, \cdots \{object_n\}$ represent unique objects in the image, where n denotes the total number of distinct objects.

OWLv2 Prompt: Fig. 9 shows the prompt for OWLv2. An example of a specific object is: *"an image of truck"*.

OWLv2

an image of {object}

Figure 9. **The Prompt of OWLv2.** The prompt of a binary classification task for OWLv2 used in all experiments, where {object} serves as a placeholder for a specific object.

B.3. Supplementary Experiments and Analysis

B.3.1. ORIC-Bench Ablation Study:

We follow the ORIC-Bench experiment settings, averaging LVLM metrics over four prompts and using a default prompt for detection models. Tab. 7 shows that both LLM-guided and CLIP-guided sampling increase question difficulty across four LVLMs and Grounding DINO Pro 1.5. LLM-guided sampling reduces yes-recall across all models, with Emu3 experiencing the largest drop (-18.50). Meanwhile, CLIP-guided sampling significantly lowers no-recall, with the most notable decline observed in DINO 1.5 Pro (-32.45). These results suggest that both positive and negative question constructions introduce challenges, though their effects differ. Notably, no-recall declines more sharply in most models. This discrepancy arises because positive questions reference real objects, aiding recognition even in incongruous backgrounds, whereas negative questions involve absent objects, leading models to over-rely on background context and hallucinate in congruous settings.

Model	Random	Pos Only	Neg Only
DINO 1.5 Pro	95.50 / 85.50	91.60 (-3.90)	53.05 **(-32.45)**
GPT-5-2025-08-07	81.53 / 96.12	71.92 (-9.61)	84.45 (-11.67)
Emu3	67.25 / 97.30	48.75 **(-18.50)**	81.17 (-16.13)
InternVL3-9B	80.88 / 97.83	68.83 (-12.05)	81.75 (-16.08)
Qwen3-VL-8B-Instruct	82.95 / 97.15	74.28 (-8.67)	83.90 (-13.25)

Table 7. **Ablation study of ORIC-Bench.** The table evaluates three sampling setups: **Random:** A baseline using randomly selected positive and negative objects. **Pos Only:** Employs LLM-guided sampling for positives and random negatives. **Neg Only:** Uses CLIP-guided sampling for negatives and random positives. All values are reported as (yes-recall / no-recall), with parentheses indicating the performance drop relative to the Random baseline.

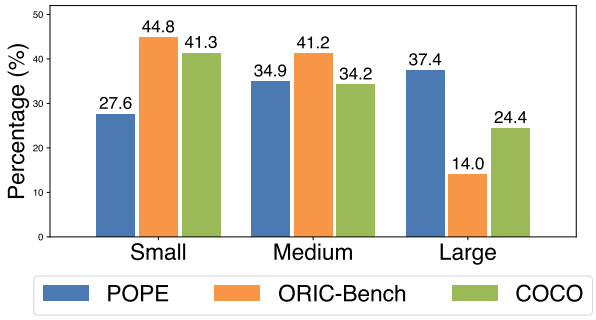


Figure 10. **Object Size Distribution across POPE, ORIC-Bench, and COCO.** Percentage distribution of small ($< 24 \times 24$ pt^2), medium (24×24–96×96 pt^2), and large ($\geq 96 \times 96$ pt^2) objects in the POPE, ORIC-Bench, and COCO datasets, highlighting ORIC-Bench's deliberate shift toward smaller and medium object scales.

B.3.2. Comparison of Object Size Distribution between POPE, ORIC-Bench, and COCO:

Fig. 10 compares the proportions of small ($< 24 \times 24$ pt^2), medium (24×24–96×96 pt^2), and large ($\geq 96 \times 96$ pt^2) objects in POPE, ORIC-Bench, and COCO. In ORIC-Bench, small objects are the single largest category at 44.8%—yet they do not constitute a majority: medium objects follow closely at 41.2%, while large objects still make up a substantial 14.0%. Relative to POPE (27.6% small, 34.9% medium, 37.4% large) and COCO (41.3% small, 34.2% medium, 24.4% large), ORIC-Bench deliberately boosts the share of small and medium instances at the expense of large ones. This design amplifies the need for fine-grained recognition and scale-robust feature extraction in the face of context incongruity, while still retaining a substantial number of medium and large objects to ensure the benchmark is not solely focused on small instances and can assess model performance across the full spectrum of object scales.

Model	POPE				ORIC-Bench			
	Precision	Recall	F1 Score	YP (%)	Precision	Recall	F1 Score	YP (%)
Closed-source								
GPT-5-2025-08-07	89.06	88.60	**88.56**	44.62	79.50	78.75	**78.61**	42.12
Encoder-based								
Llama-3.2-11B-Vision	25.00	50.00	33.33	0.00	25.00	50.00	33.33	0.00
Chameleon-7B	47.08	50.01	33.95	99.29	59.75	50.10	34.08	99.28
BLIP-3	36.20	44.88	37.29	80.30	43.14	49.86	42.99	81.54
VILA1.5-13B	60.87	59.92	57.49	36.80	65.19	62.40	60.41	28.95
GLM-4v-9B	86.55	84.12	83.85	37.30	71.18	64.92	61.99	23.32
Phi-3.5-Vision-Instruct	86.76	86.28	86.23	44.35	68.69	68.06	67.79	40.86
InternLM-XComposer2.5-7B	84.72	83.16	82.98	39.84	73.32	70.35	69.33	33.77
SmolVLM2-2.2B-Instruct	87.57	86.89	86.83	43.56	72.87	71.44	70.95	38.01
Kimi-VL-A3B-Instruct	88.91	87.69	87.59	41.19	74.67	72.28	71.58	34.45
Molmo-7B-D-0924	83.76	81.45	81.03	61.42	78.92	73.74	71.95	69.34
LLaVA-v1.6-Vicuna-13B	88.24	88.14	88.13	51.39	75.29	74.56	74.37	56.94
Janus-Pro-7B	87.32	87.03	87.00	50.65	76.60	75.22	74.83	56.42
InternVL3-9B	88.8	88.69	**88.68**	47.96	77.33	76.95	76.87	44.60
Qwen3-VL-8B-Instruct	88.13	88.04	88.03	47.66	79.93	79.61	**79.55**	44.94
Encoder-free								
Fuyu-8B	68.39	53.47	40.48	95.70	44.83	50.16	34.16	99.29
EVE-7B-HD-v1.0	82.19	79.81	79.34	61.36	61.02	56.42	51.59	76.53
Emu3-Chat	87.43	86.72	**86.66**	43.25	67.74	65.79	**64.78**	33.41
Open-vocabulary Detection								
OWLv2	86.74	86.55	**86.53**	53.55	73.02	72.25	72.02	40.85
Grounding DINO 1.5 Pro	85.62	85.05	84.99	56.35	77.02	73.40	**72.48**	68.30

Table 8. **Full Model Performance Comparison: POPE vs. ORIC.** The table compares POPE and ORIC across various model categories: closed-source, encoder-based, encoder-free, and open-vocabulary detection models. Performance is evaluated using macro precision, recall, and F1 score. The yes proportion (YP (%)) indicates the percentage of "yes" predictions. "Prec." denotes precision, "Rec." denotes recall, and "F1." denotes the F1 score. All values are averaged across four prompts, except for detection models, which use a single prompt without averaging.

B.3.3. Full Results of Comparison between POPE and ORIC

Tab. 8 presents a comparative analysis of POPE and ORIC-Bench across 19 LVLMs and 2 open-vocabulary detection models. Notably, the macro F1 scores of Llama-3.2-11B-Vision, Chameleon-7B, BLIP-3, and VILA1.5-3B in POPE are comparable to or even exceed those in ORIC-Bench. A potential explanation is that these models exhibit a high proportion of "yes" responses in both benchmarks, suggesting a tendency to answer affirmatively regardless of context. This behavior indicates limited object recognition capabilities, as their responses remain consistent across different evaluation settings. Furthermore, the macro precision and recall of other models in ORIC-Bench are significantly lower than in POPE, leading to a sharp decline in macro F1 scores. This suggests that ORIC-Bench presents a greater challenge for all tested LVLMs, highlighting their struggles with object recognition, particularly when considering contextual incongruity.

B.3.4. Threshold Analysis of Open-vocabulary Detection Models

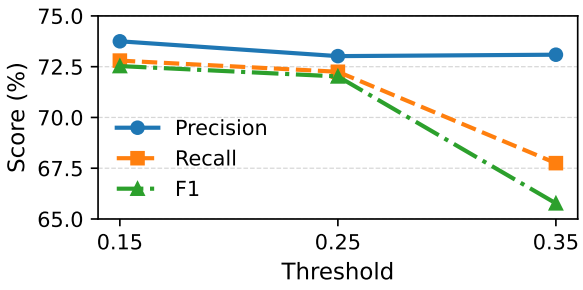


Figure 11. **OWLv2 Threshold Sweep.** Macro precision, recall, and F1 score versus the decision threshold of OWLv2.

For open-vocabulary detection with OWLv2, we conduct a threshold sweep to determine the optimal decision

boundary for object presence. As shown in Fig. 11, thresholds of 0.15 and 0.25 yield similar macro precision (∼73%) and comparable F1 scores, while increasing to 0.35 reduces recall and F1 due to overly conservative predictions. We therefore adopt 0.25 as the default threshold, balancing precision and recall for reliable evaluation under contextual incongruity.

B.3.5. Mechanistic Error Analysis

To understand persistent failure modes, we conduct a mechanistic analysis on prompt-robust failures. For each of the top three models (Qwen3-VL-8B, GPT-5-2025-08-07, and InternVL3-9B), we analyze 100 cases that remain incorrect across all four prompt variants, categorizing them by contextual incongruity type: *scene-level*, where incongruity is driven by global scene semantics; *co-occurrence*, based on the typical co-presence of objects; and *set-completion*, where visible objects nearly form a functional set and the queried object is the missing component. As shown in Tab. 9, scene-level errors dominate across all models (23–25 cases), followed by co-occurrence (14–19) and set-completion (8–11). This consistent pattern suggests that global scene semantics pose the greatest challenge for current LVLMs, as models tend to over-rely on holistic scene priors when local object evidence is ambiguous.

Model	Scene-level	Co-occurrence	Set-completion
Qwen3-VL-8B-Instruct	23	18	9
GPT-5-2025-08-07	25	14	11
InternVL3-9B	23	19	8

Table 9. **Prompt-Robust Error Analysis.** Error distribution across contextual categories for cases that remain incorrect under all prompts (100 cases per model). The contextual categories include scene-level expectations driven by global scene semantics, co-occurrence based on the typical co-presence of objects, and set-completion, where visible objects nearly form a functional set and the queried object is the missing component.

C. Visual-RFT Experimental Details

C.1. R1-Style Prompt for Reinforcement Fine-Tuning

Fig. 12 shows the R1-style prompt used in our reinforcement fine-tuning (RFT) experiments. An example of a specific question is: *"Is there a cat in the image?"*.

C.2. Zero-Shot CoT Prompt of LVLMs:

Fig. 13 shows the zero-shot CoT prompt for LVLMs. An example of a specific question is: *"Is there a person in the image?"*.

R1-Style Prompt for Visual RFT

<image>
Prompt: Is there a/an {object} in the image? Please first provide your reasoning or working out on how you would go about solving the question between <REASONING> and </REASONING> and then your final answer between <SOLUTION> and (put yes or no here) </SOLUTION>.

Figure 12. **The R1-style prompt used for reinforcement fine-tuning.** The prompt elicits explicit reasoning (<REASONING>...</REASONING>) and a verifiable final answer (<SOLUTION>...</SOLUTION>) to enable reward evaluation.

Zero-Shot CoT of LVLMs

<image>
Question: {question}
Let's think step-by-step and then answer the question based on the given image.

Figure 13. **The zero-shot CoT Prompt of LVLMs.** The prompt of a binary classification task for LVLMs using zero-shot CoT prompting strategy.

C.3. Visual-RFT Training Hyper-parameters

Tab. 10 lists the full set of hyper-parameters used in our Visual-RFT training. We include all optimization, sampling, and generation settings to ensure complete reproducibility.

C.4. Evaluation Subset Details of HallusionBench and AMBER

To evaluate out-of-distribution generalization, we construct evaluation subsets from HallusionBench and AMBER. Each subset contains 200 binary questions with balanced labels (100 yes and 100 no).

HallusionBench. HallusionBench evaluates hallucinations in multimodal models using images containing visual illusions and abstract figures. We construct our subset by selecting samples from the figure and illusion subcategories. These questions require models to rely on visual evidence rather than language priors. The subset includes problems such as determining whether certain objects appear in a figure, whether all items belong to a specific category, or whether geometric properties in illusion images (e.g., relative size, length, color, or alignment of shapes) are actually the same.

Hyper-parameter	Configuration
VLM Init	Qwen3-VL-8B-Instruct
KL Penalty (β)	0
Optimizer	AdamW
Learning Rate	2×10^{-6}
Clipping Range ϵ	0.2
LR Scheduler	Cosine
Weight Decay	0
Precision	BF16
Gradient Clipping	1.0
Per-device Batch Size	1
Gradient Accumulation	4
Rollout Temperature	0.7
Rollout Top-p	0.8
Rollout Top-k	20
Group Size G	8
Max Prompt Length	1024
Max Completion Length	256
Epochs	15
GPUs	4× NVIDIA H100 80GB

Table 10. **Training Configuration.** Key hyperparameters for GRPO-based Visual-RFT of Qwen3-VL-8B-Instruct.

AMBER. AMBER evaluates hallucination through discriminative visual reasoning tasks involving existence, attribute, and relation queries. Existence questions ask whether a specific object appears in the image. Attribute questions evaluate properties such as color, state, number, or actions of objects (e.g., whether the grass is green or whether two people are present). Relation questions require reasoning about interactions or spatial relations between objects, such as whether two objects are in direct contact. These tasks collectively assess the model's ability to reason about objects, their properties, and their relationships without hallucinating unsupported visual content.

D. CLIPScore as a Proxy for Contextual Alignment

While CLIPScore is not a perfect object detector and has known limitations in capturing compositional semantics [31, 83], we use it solely as an external probe to assess the contextual alignment of replaced objects. Specifically, CLIP-guided sampling is applied only to "no"-label cases to select ground-truth nonexistent yet contextually plausible objects with higher CLIPScores, thereby constructing more challenging negatives. Our ablation study B.3 confirms this strategy by showing a significant reduction in negative recall, indicating increased contextual incongruity.

Importantly, CLIPScore is never used for model evaluation but serves as a heuristic signal of object–context compatibility. To ensure robustness, we validate our findings across three independent CLIP variants in A.3, all consistently showing that ORIC "yes" or "no" pairs exhibit higher visual similarity than those in POPE, thus increasing task difficulty. While CLIP's co-occurrence bias may contribute to high scores for out-of-context objects, we argue that this reflects its tendency to associate such objects with plausible scenes, which is precisely the type of confounding signal our benchmark aims to capture. Despite these limitations, CLIPScore remains a useful proxy for semantic alignment, as supported by recent work [28, 76].

E. Visualization of ORIC-Bench Examples and Analysis

E.1. Human Evaluation Details

Incongruity Strength	Label Yes	Label No
Strong Incongruity	82	87
Borderline Incongruity	65	60
Annotation Error	3	3

Contextual Category	Label Yes	Label No
Scene-level	70	75
Co-occurrence	61	58
Set-completion	16	18

Table 11. **Top.** Distribution of Yes/No labels across different levels of contextual incongruity. **Bottom.** Distribution of labels across contextual incongruity categories, including scene-level expectations driven by global scene semantics, co-occurrence based on typical co-presence of a few objects, and set-completion, where visible objects nearly form a functional set, and the queried object is the missing component.

We sampled 150 "yes" and 150 "no" questions generated by the ORIC framework and manually verified two aspects: object labeling accuracy and contextual incongruity. As shown in the top of Tab. 11, 147 of the 150 "yes" examples are strong or borderline incongruous, while 147 of the 150 "no" examples are contextually expected, with only three annotation errors. The bottom of Tab. 11 further categorizes questions into three contextual types: scene-level, co-occurrence, and set-completion, as illustrated in Appendix B.3.5. The distribution shows that ORIC generates challenging questions across diverse contextual incongruity patterns.

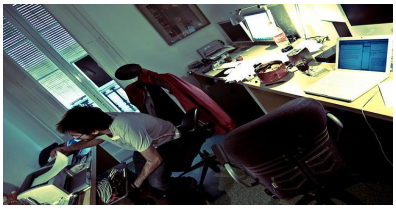
E.2. Error Questions from Human Evaluation

Fig. 14 shows six error cases from the 300 sampled questions (150 "yes" and 150 "no") in ORIC-Bench based on the MSCOCO dataset. We examine two aspects: object labeling accuracy and whether the visual context creates the intended incongruity. The errors fall into two categories:

- **Inaccurate Object Labeling**: The annotated object does not match the actual image content due to errors in the MSCOCO annotations.
- **Insufficient Contextual Incongruity**: The visual context does not create a clear incongruity. In some "yes" questions, the context remains compatible with the target object, while in some "no" questions, the context fails to create a meaningful contradiction.

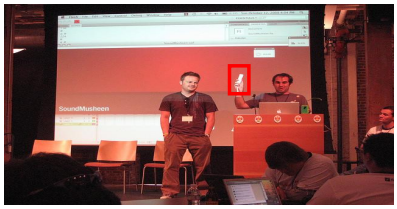
E.3. ORIC Question Examples

Fig. 15 presents various examples from ORIC. In "yes"-label and "no"-label questions, visual contexts are incongruous with the question-related objects. Our LLM-guided and CLIP-guided sampling method effectively generates challenging questions considering contextual incongruity.



POPE: Is there a keyboard in the image?
Label : No
Inaccurate Object Labeling: The keyboard is present while labeling errors.

Question: Is there a mouse in the image?
Label: Yes
Not Causing The Incongruous Context: The office area provides a congruous context for a mouse.



Question: Is there a remote in the image?
Label: Yes
Not Causing The Incongruous Context: The conference room provides a congruous context for a remote.

Question: Is there a bed in the image?
Label: No
Not Causing The Incongruous Context: The living room doesn't provide an incongruous context for a nonexistent bed.

Question: Is there an orange in the image?
Label: No
Not Causing The Incongruous Context: The tennis court doesn't provide an incongruous context for a nonexistent orange .

Question: Is there a skateboard in the image?
Label: Yes
Not Causing The Incongruous Context: The skatepark doesn't provide an incongruous context for a nonexistent skateboard.

Figure 14. **Error Examples of ORIC from Human Evaluation.** There are six error cases among the **300** sampled questions in ORIC using the MSCOCO dataset, resulting in an error rate of 2%. These errors can be classified into two categories. **Inaccurate Object Labeling** occurs when the labeled object's presence does not match the actual content of the image. **Not Causing the Incongruous Background** includes cases where the visual context aligns with an existent object in a "yes"-label question or does not introduce incongruity for a nonexistent object in a "no"-label question.

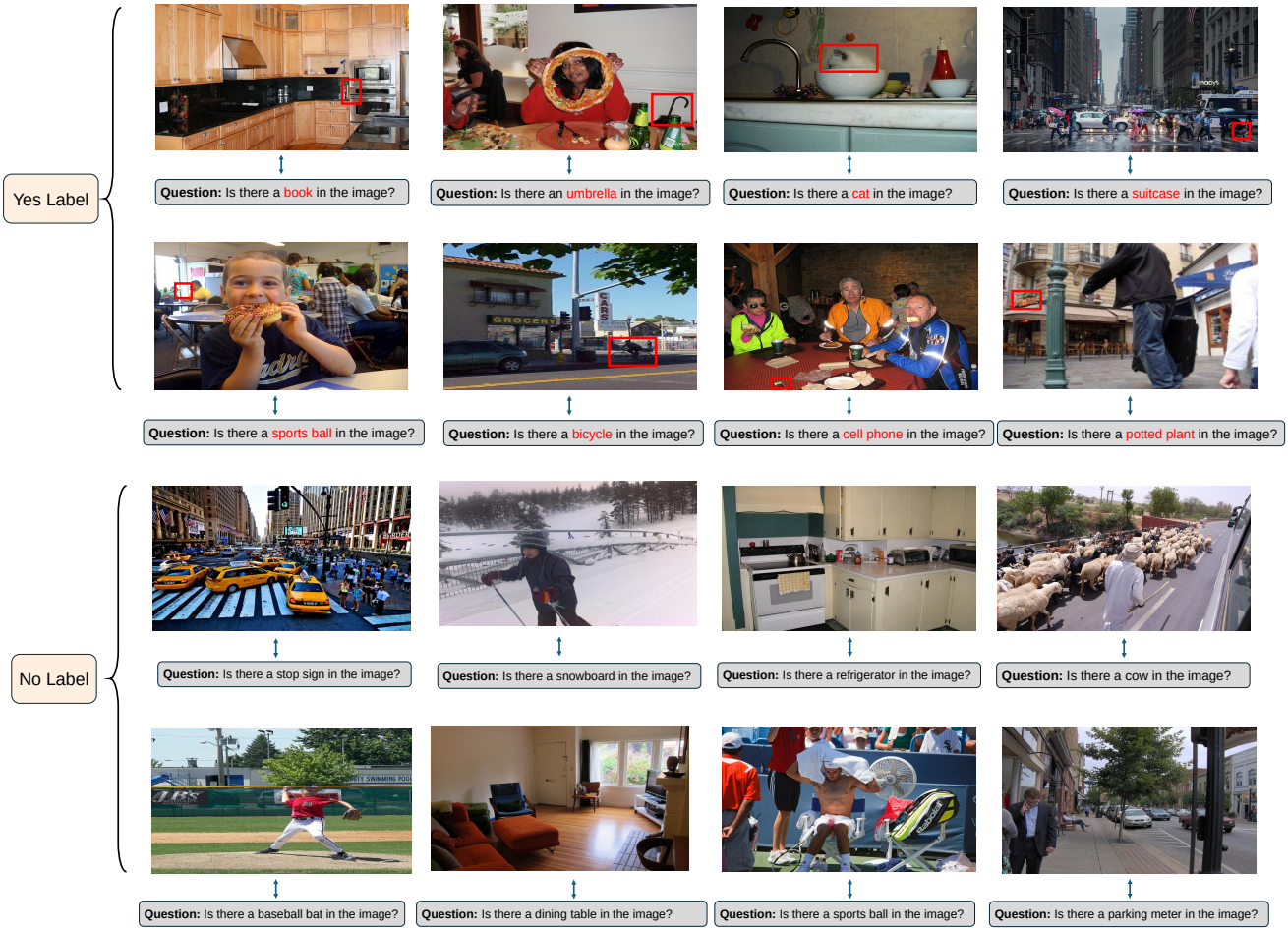


Figure 15. **Question Examples of ORIC.** The figure shows sampled question examples from ORIC using the MSCOCO dataset. The first and second rows contain questions labeled "yes," while the third and fourth rows contain questions labeled "no." The red box highlights the bounding boxes of existing objects in "yes"-label questions.

Model	Overall				Label Yes			Label No		
	Pre.	Rec.	F1	YP (%)	Pre.	Rec.	F1	Pre.	Rec.	F1
Closed-source										
GPT-5-2025-08-07	79.50	78.75	**78.61**	42.12	84.14	70.88	**76.92**	71.84	88.62	**79.35**
Vision-encoder-based										
Llama-3.2-11B-Vision	25.00	50.00	33.33	0.00	0.00	0.00	0.00	50.00	100.00	66.67
Chameleon-7B	59.75	50.10	34.08	99.28	50.05	99.38	66.57	69.45	0.82	1.59
BLIP-3	43.14	49.86	42.99	81.54	45.36	51.22	47.02	40.92	48.50	38.96
VILA1.5-13B	65.19	62.40	60.41	28.95	71.44	41.35	51.86	58.92	83.45	68.96
GLM-4v-9B	71.18	64.92	61.99	23.32	82.41	38.25	51.61	59.94	91.60	72.35
Phi-3.5-Vision-Instruct	68.69	68.06	67.79	40.86	72.12	58.92	64.85	65.27	77.20	70.73
InternLM-XComposer2.5-7B	73.32	70.35	69.33	33.77	80.96	54.12	64.17	65.67	86.58	74.49
SmolVLM2-2.2B-Instruct	72.87	71.44	70.95	38.01	78.30	59.45	67.38	67.44	83.42	74.52
Kimi-VL-A3B-Instruct	74.67	72.28	71.58	34.45	82.32	56.73	67.13	67.02	87.83	76.02
Molmo-7B-D-0924	78.92	73.74	71.95	69.34	68.22	93.08	76.61	89.62	54.40	65.59
LLaVA-v1.6-Vicuna-13B	75.29	74.56	74.37	56.94	71.76	81.50	76.19	78.82	67.62	72.55
Janus-Pro-7B	76.60	75.22	74.83	56.42	73.30	81.65	76.71	79.90	68.80	72.95
InternVL3-9B	77.33	76.95	76.87	44.60	80.27	71.55	75.60	74.39	82.35	78.13
Qwen3-VL-8B-Instruct	79.93	79.61	**79.55**	44.94	82.96	74.55	**78.51**	76.91	84.68	**80.59**
Vision-encoder-free										
Fuyu-8B	44.83	50.16	34.16	99.29	50.08	99.45	**66.61**	39.59	0.88	1.71
EVE-7B-HD-v1.0	61.02	56.42	51.59	76.53	54.82	82.95	65.27	67.22	29.90	37.90
Emu3-Chat	67.74	65.79	**64.78**	33.41	73.58	49.20	58.90	61.91	82.38	**70.67**
Open-vocabulary Detection										
OWLv2	73.02	72.25	72.02	40.85	77.23	63.10	69.46	68.81	81.40	**74.58**
Grounding DINO 1.5 Pro	77.02	73.40	**72.48**	68.30	67.13	91.70	**77.51**	86.91	55.10	67.44

Table 12. **Full Experimental Results on ORIC-Bench.** Performance is broken down by model category and label type (Yes/No). We report macro precision (Prec.), recall (Rec.), F1 score, and the proportion of "yes" predictions (YP). Results for LVLMs are averaged over four prompts, while detection models use a single prompt.