DeepScan: A Training-Free Framework for Visually Grounded Reasoning in Large Vision-Language Models

Supplementary Material

Outline

§ A **Discussions** includes the *failure cases* of DeepScan, limitations and future work, and Broader Impacts, delivering valuable insights.

§ B **Methodological Details** provides pseudocode, the exact prompts applied, and hyperparameter settings.

§ C **Additional Results** conducts a comparison against strong baselines and reveals the state-of-the-art performance achieved by DeepScan on fine-grained tasks.

§ D **Additional Cases** including the advantage of the *bottom-up* grounding paradigm and the Example Model Outputs

A. Discussions

Failure Cases. Based on our analysis, DeepScan's failures mainly fall into two categories.
(1) Grounding failure. In direct-attribute recognition, when multiple visually similar objects remain within the evidence neighborhood, the experts may propose incorrect evidence and the LVLM may misjudge it, leading to an incorrect evidence localization; and thus the LVLM produces an incorrect answer based on wrong evidence (see Fig. 12).
(2) Reasoning failure. For spatial-relation reasoning with multiple pieces of evidence, DeepScan currently forms a merged evidence view via the minimal enclosing bounding box over all localized evidence. When the evidence is widely separated, this large crop introduces substantial inter-evidence noisy context that hampers reasoning; the issue is exacerbated when the merged view includes distracting content that conflicts with the correct answer (see Fig. 13). This points to a promising direction: replace the minimal enclosing box with a generative composition that reassembles localized fine-grained evidence into a compact layout, suppressing inter-evidence context while preserving object attributes and spatial relations.

Limitations. Compared with one-shot evidence-detection pipelines for visually grounded reasoning, DeepScan has higher inference latency. Nevertheless, as a test-time scaling paradigm, DeepScan offers a controllable performance-efficiency trade-off by tuning the patch size and the number of evidence proposals. In practice, its overall performance-efficiency profile remains acceptable relative to advanced visually grounded models (*e.g.*, GPT-5), as detailed in §D (Example Model Outputs). DeepScan's current implementation sets a relatively small patch size per image to capture fine-grained cues, which is unnecessary and inefficient for simple cases with salient evidence. In the future, we plan to develop an adaptive partition strategy that flexibly assigns a specific patch size within each grounding process based on stronger priors (*e.g.*, evidence saliency), which is expected to further improve the performance-efficiency trade-off.

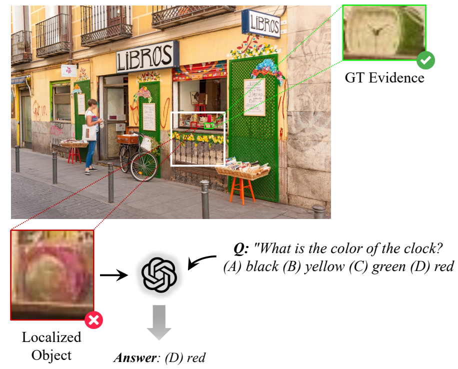


Figure 12. Case study of grounding failure

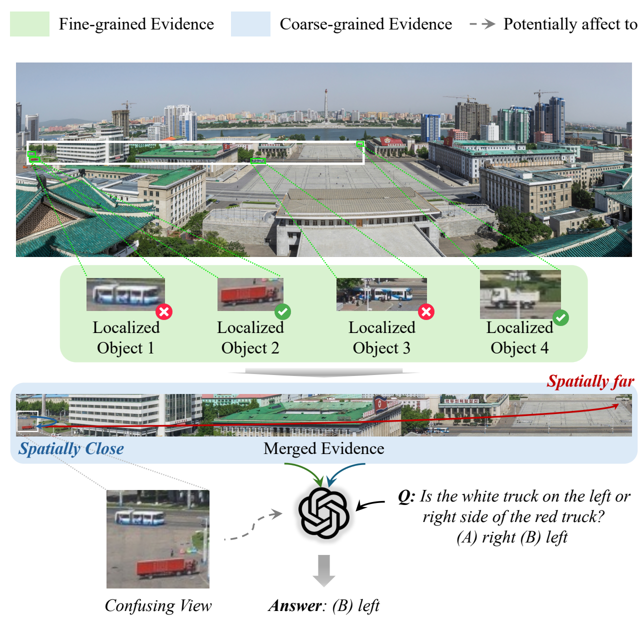


Figure 13. Case study of reasoning failure

Broader Impacts. We propose DeepScan, a training-free visually grounded reasoning framework that replaces brittle one-shot localization with hierarchical scanning and refocusing. By tying answers to concrete visual evidence and making contextual extent explicit, DeepScan can benefit domains needing reliable, fine-grained perception under clut-

System Prompt

You are an advanced image understanding assistant. You will be given an image and a question about it.

User Prompt 1 (Evidence Decomposition)

Task: List objects mentioned in text in List format.
Input text: {question}
Action: What objects are mentioned in original text? List separated by commas. For example, from "person with white trousers on the left or right side of the person in blue", output "["person with white trousers", "person in blue"]".

User Prompt 2 (Evidence Judgment)

I will provide you an image and a **question**: {question}, please firstly determine whether the image contains the clues for answering the question or not (answer with **Yes** or **No**); then give the evidence of your decision.

User Prompt 3 (View Completeness Justification)

Question: Does the image fully contain every object in the list {target_list}? Please treat "fully contain" as entirely within the frame (not truncated by image boundaries). Please firstly answer the question with **Yes** or **No**; then give the evidence of your decision. For example, if yes, list the evidence of each object (*e.g.*, object: bbox [x1, y1, x2, y2] or a clear region description); if no, list the missing objects by name.

Algorithm 1 Hierarchical Scanning *with* Acceleration

Require: image $I \in \mathbb{R}^{H\times W\times 3}$, question q, patch size $\{l_s, l_c\}$, candidate count k, kernel $\mathcal{K}$, $\mathcal{S}_r$, threshold $\tau_{\text{area}}, \theta_{\text{IoU}}$
Ensure: Evidence set $\mathcal{E} = \{(b_i, e_i)\}$

$\mathcal{E}_{\text{all}} \leftarrow \emptyset, \quad \mathcal{E} \leftarrow \emptyset$
$l \leftarrow \textsc{SelectByLen}(\textsc{Lvlm}(\texttt{Prompt1}(q)), \{l_s, l_c\})$
for each patch p in $\textsc{Partition}(I, l)$ **do**
  $S_p \leftarrow \textsc{Search}(p, q)$
  $S_p^+ \leftarrow \mathbb{I}(S_p \geq \text{OTSU}(S_p))$
  for each $G \in \textsc{ConnComp}(S_p^+)$ with $|G| \geq \tau$ **do**
    $d_G(i,j) \leftarrow \inf_{\gamma\in\partial G} \|(i,j) - \gamma\|_2, \ \forall (i,j) \in G$
    $\tilde{S}_p \leftarrow \textsc{Norm}(S_p), \quad \tilde{d} \leftarrow \textsc{Norm}(d_G)$
    $c^\star \leftarrow \arg\max_{c\in G} \tilde{S}_p(c) \cdot \tilde{d}(c)$
    $c' \leftarrow \textsc{LiftToImage}(c^\star, p \rightarrow I), \quad \mathcal{C}_p' \cup \{c'\}$
while $\mathcal{C}_p'$ not empty **do**
  $c \leftarrow \textsc{Pop}(\mathcal{C}_p), \quad m \leftarrow \textsc{Segment}(I, c)$
  $m^+ \leftarrow (\, m \bullet \mathcal{K}\,) \oplus \mathcal{S}_r$
  $b \leftarrow \textsc{BBox}(m^+), \quad e \leftarrow \textsc{Crop}(I, b)$
  if $\text{IoU}(b, b_i) \leq \theta_{\text{IoU}}$ for all $(b_i, e_i) \in \mathcal{E}_{\text{all}}$ **then**
    $\mathcal{E}_{\text{all}} \leftarrow \mathcal{E}_{\text{all}} \cup \{(b, e)\}$
  $I \leftarrow I \odot (\mathbf{1} - m^+), \ \mathcal{C}_p' \leftarrow \{c' \in \mathcal{C}_p' \mid m^+(c') = 0\}$
$\mathcal{E}_k \leftarrow \textsc{TakeKSmallestByArea}(\mathcal{E}_{\text{all}}, k)$
for each $(b, e) \in \mathcal{E}_k$ **do**
  if $\textsc{Lvlm}(e, \texttt{Prompt2}(q)) = \texttt{Yes}$ **then**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(b, e)\}$
return $\mathcal{E}$

Algorithm 2 Refocusing

Require: image $I \in \mathbb{R}^{H\times W\times 3}$, question q, target list t, evidence set $\mathcal{E}$
Ensure: refined view V

$\mathcal{R} \leftarrow \emptyset, \quad b_{\text{m}} \leftarrow \bigcup_{(b,e)\in\mathcal{E}} b$
$e_{\text{m}} \leftarrow \textsc{Crop}(I, b_{\text{m}}), \quad V_1 \leftarrow e_m$
$V_2 \leftarrow \textsc{In}(V_1, q), \quad V_3 \leftarrow \textsc{Out}(V_1), \quad V_4 \leftarrow \textsc{In}(V_3, q)$
for $V \in \{V_1, V_2, V_3, V_4\}$ **do**
  if $\textsc{Lvlm}(V, \texttt{UserPrompt3}(t)) = \texttt{Yes}$ **then**
    $h, w \leftarrow \textsc{Shape}(V)$
    $R \leftarrow HW/hw$
  else
    $R \leftarrow 0$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$
$i^* \leftarrow \arg\max_{i\in\{1,2,3,4\}} \mathcal{R}(i)$
return V_{i^*}

ter and occlusion. Examples include GUI agents (anchoring actions to the correct widget, explaining clicks), embodied manipulation (grasping small parts), and autonomous driving (partially occluded signs). Potential risks include propagation of biases inherited from LVLMs and experts, and automation errors in safety-critical settings. Besides, as a test-time scaling paradigm, DeepScan introduces extra inference overhead, which may limit deployment in compute-constrained scenarios such as edge or mobile devices.

B. Methodological Details

Prompts. We provide below the prompts involved in Sec. 3: (i) *Evidence Decomposition* (to set the patch size), (ii) *Evidence Judgment* in *Hierarchical Scanning*, and (iii) *View Completeness Justification* in *Refocusing*.

Hyper-Parameters. In *Local Cue Exploration*, the area threshold for noisy-cue filtering is set to 50 pixels. In *Multi-Scale Evidence Extraction*, morphological post-processing uses a 5×5 flat structuring element $\mathcal{K}$ and a disk $\mathcal{S}_r$ with ra-

Table 8. Comparisons between the proposed DeepScan and existing visually grounded reasoning approaches. Like most test-time scaling paradigms, DeepScan is easier to scale up. Furthermore, DeepScan introduces hierarchical scanning and refocusing for a robust bottom-up evidence localization and recalibration and leverages hybrid granular evidence to enable LVLM to produce higher-quality answers.

Methods	Venue	Ease to Scaling	External Experts	Search Strategy	Grounding Paradigm	Evidence Granularity	Inference Latency
Seal [45]	CVPR'24	✗	✗	LLM-guided Search	Coarse-to-Fine	Fine	High
Dyfo [13]	CVPR'25	✓	✓	Detection + MCTS	Coarse-to-Fine	Coarse	High
DeepEyes [53]	NeurIPS'25	✗	✗	Generative Bbox	Coarse-to-Fine	Fine	High
PixelReasoner [31]	NeurIPS'25	✗	✗	Generative Bbox	Coarse-to-Fine	Coarse	High
ViGoRL [28]	NeurIPS'25	✗	✗	Generative Bbox	Coarse-to-Fine	Fine	High
ZoomRefine [50]	NeurIPS'25	✓	✗	Generative Bbox	Coarse-to-Fine	Coarse	Medium
TreeVGR [35]	Preprint, Jul	✗	✗	Generative Bbox	Implicit Search	NA	Low
Thyme-VL [51]	Preprint, Aug	✗	✗	Generative Code	Coarse-to-Fine	Coarse	High
DeepScan (Ours)	–	✓	✓	Hierarchical Scanning + Refocusing	Bottom-up	Hybrid	High

dius r=20. The IoU threshold for filtering similar evidence is θ_{IoU}=0.3. We set k=10, *i.e.*, retain only the 10 smallest pairs $(b, e) \in \mathcal{E}$ for acceleration. For *Refocusing*, to prevent undersized crops, we pad the visual expert's detections ($\mathcal{B}$) by 28 pixels on all sides, and set the scaling factor to s=1.5. For *LVLM Querying*, the maximum output length is 50 for *Evidence Decomposition*, *Evidence Judgment*, and *View Completeness Justification*, and 1024 for *Evidence-Enhanced Reasoning*. During inference, we use temperature t=0 with a fixed random seed (13). Beam search and top-k sampling are disabled by default.

Pseudocodes. Algorithmic details for Hierarchical Scanning and Refocusing are provided in Algorithms 1 and 2. Here, the prompts `Prompt1`, `Prompt2`, and `Prompt3` are specified in the **Prompt** subsection, *i.e.*, "User Prompt 1-3"; `Prompt1`(x) denotes instantiating the prompt with the string x via template-based substitution. In addition, LIFTTOIMAGE maps patch coordinates to the full-image coordinate system. CLOSE and DILATE indicate morphological closing and dilation, respectively. The operator $\odot$ denotes element-wise multiplication.

C. Additional Results

Baselines. SEAL [45], PixelReasoner [31], TreeVGR [35], DeepEyes [53], ViGoRL [28], and Thyme-VL [51] depend on learned localization modules or RL-based decision controllers for search. Upgrading these methods to a stronger LVLM typically requires retraining these components. In contrast, DyFo [13] performs MCTS over focus actions with a visual expert, and ZoomRefine [50] provides an unlearning variant that leverages LVLM prior for visually grounded reasoning through prompt engineering; both scale more readily across larger LVLM backbones. Our method is easier to scale up and introduces a novel bottom-up hierarchical scanning and refocusing for context-optimal views, complemented by a hybrid evidence memory, yielding better robustness than existing approaches.

Table 9. Additional comparison results on V* Bench between our method with existing baselines. We produced the results† with the provided official codes for fair comparisons.

	Overall	Attribute	Spatial
o3	95.0	-	-
Seal	74.8	76.3	75.4
Dyfo-L	62.7	53.9	59.2
Dyfo-Q	80.0	82.9	81.2
Qwen2.5-VL-7B	74.3	77.4	69.7
PixelReasoner	80.6	83.5	76.3
Thyme-VL	82.2	83.5	80.3
ZoomRefine†	82.2	85.3	77.6
Dyfo†	84.3	82.6	86.8
TreeVGR†	85.9	86.1	85.5
ViGoRL	86.4	-	-
DeepEyes	90.0	92.1	86.8
DeepScan ($k = 10$)	90.6	93.0	86.8
DeepScan ($k = \infty$)	91.1	93.9	86.8
Qwen2.5-VL-72B	84.8	90.8	80.9
DeepScan-72B ($k = 10$)	93.7	93.9	93.4
DeepScan-72B ($k = \infty$)	94.2	94.8	93.4

Performance Comparison. As shown in Table 9, **DeepScan** attains **90.6**% at k=10 and **91.1**% at k=∞ based on Qwen2.5-VL-7B, which substantially outperforms both the training-free ZoomRefine† (82.2%) and DyFo† (84.3%) and even surpasses the RL-based DeepEyes (90.0%), ViGoRL (86.4%), TreeVGR (85.9%), Thyme-VL (82.2%), and PixelReasoner (80.6%) on the V* Benchmark. Specifically, on the *Attribute* subset, DeepScan reaches 93.0-93.9% while maintaining *Spatial* at 86.8%, indicating that bottom-up recovery plus refocusing preserves necessary

Figure 14. Qualitative analysis of the grounding paradigms with the attention map S of the *search expert*

context for reasoning. When scaling the LVLM backbone to 72B, **DeepScan-72B** achieves **94.2**% overall and 93.4% on the *Spatial* subset, a 9.4% gain over the Qwen2.5-VL-72B (84.8%). These results demonstrate DeepScan as a training-free framework that consistently improves diverse LVLMs and benefits from model scaling, with a controllable performance-latency trade-off via top-k selection.

Engineering Optimization. Our standard implementation of DeepScan relies on nested loops using the *Hugging face transformers backend* (as detailed in Algorithms 1 and 2). However, this implementation is severely bottlenecked by inefficient **sequential computation** during the Scanning and Refocusing stages. Furthermore, the frequent communication overhead between the visual expert and the LVLM leads to poor GPU utilization. Compounded by the *lack of modern primitives* in the standard **HF Transformers backend**, these factors result in substantial inference latency (as shown in Fig. 7). To address this, we introduce a suite of engineering optimizations that exploit DeepScan's inherent algorithmic properties, significantly enhancing its viability for latency-sensitive applications. Specifically, unlike MCTS-based methods (*e.g.*, Dyfo), DeepScan is built on *deterministic sampling*, enabling it to fully benefit from parallel acceleration via batching. Hence, we implemented batch processing for **(a)** attention map calculation, **(b)** top-k candidate judgment, and **(c)** evidence view justification, and further parallelized post-processing routines to reduce tail latency. This strategy compresses many sequential in-

	Qwen2.5-VL 7B	DeepEyes	Zoom Refine	Dyfo	Hierarchical Scan	Refocus	DeepScan
Accuracy (%)	75.4	89.0	82.7	83.8	84.8	89.5	90.1
Token Cost (k)	3.5	13	5.1	7.2	6.5	8.1	8.4
End-to-End Latency (s)	0.4	6.9	0.9	2.4	2.2	3.0	3.1

Table 10. Performance–efficiency comparison on V* Benchmark. *End-to-End* Latency is the wall-clock time *per sample* from input to final output, *including all auxiliary steps* (*e.g.*, Evidence Decompositions, Expert Calls, Post-Processing, Evidence Judgments).

teractions into a *single batched* search expert call and *three* LVLM forward passes *with extended context*, thus *substantially improving GPU utilization and reducing GPU idle time*. Moreover, we migrated to the *vLLM backend*, leveraging PagedAttention and optimized CUDA kernels. *Collectively, these standard engineering optimizations yield an* $\sim 8\times$ ***speedup*** *over our previous implementation.*

Performance-Latency Trade-off. Integrating the above optimizations, we re-evaluated DeepScan via *VLMEvalKit* (with vLLM backend) on 4×L20 GPUs. As shown in Table 10, compared to *DeepEyes* that relies on multi-turn tool executions, DeepScan exhibits striking superiority. Specifically, DeepEyes suffers a severe 6.9s latency and 13k token cost, while DeepScan attains higher accuracy (90.1% *vs.* 89.0%) with ~2.2× faster speed (3.1s) as well as ~35% fewer tokens. This reveals that batched deterministic sampling effectively bypasses the sequential overhead of agentic paradigms, yielding superior reasoning with higher efficiency. Against the MCTS-based method *Dyfo*, DeepScan also offers a favorable trade-off. It incurs a marginal latency overhead (3.1s *vs.* 2.4s) but delivers a substantial +6.3% accuracy gain. Notably, our *Hierarchical Scan* alone outperforms Dyfo (84.8% *vs.* 83.8%) with lower latency (2.2s).

In fine-grained scenarios, the target occupies a minuscule fraction of the image, resulting in an inherently low signal-to-noise ratio (SNR) where the effective visual signal is overwhelmed by context noise. In these cases, search-based methods struggle to identify a reliable initial anchor, frequently causing exhaustive tree expansion to degenerate into an inefficient random walk. By slicing the image and evaluating patches via a single batched forward pass, our hierarchical scan explicitly isolates the target, drastically amplifying the local SNR. Coupled with the subsequent *Refocus*—which requires merely 0.8s but yields a massive +4.7% performance leap—these results reveal a profound insight: systematically scaling compute to deterministically maximize visual SNR is fundamentally more efficient than heuristic search-tree expansion.

D. Additional Visualization

D.1. Qualitative Analysis of Grounding Paradigm

Direct Attributes. For attribute questions (*e.g.*, helmet/slippers/candles color, dog breed), the *one-shot* variant tends to lock onto globally salient but irrelevant regions, a typical *attention sink/drift* failure, yielding mis-localization or missing the true fine-grained cue. By contrast, our *bottom-up* paradigm begins with local cue exploration and iteratively recenters the view on the true evidence, effectively "pulling" attention back to the correct object—consistent with the figure's transitions from *Attention Sink* to *Align to* on attribute examples. This behavior aligns with the paper's analysis that top-down, one-shot localization is fragile under noisy context, whereas bottom-up scanning suppresses distractions and aligns attention with the correct target.

Spatial Relations. Relation questions (*e.g.*, left/right of motorcyclist vs. truck, sky wheel vs. rider, blue luggage vs. bus) require jointly grounding *two* targets. The one-shot variant often fixates on a single salient object and *misses the subtle counterpart*, leading to incorrect relational judgments; bottom-up scanning progressively resolves and aligns to *both* entities, producing reliable spatial decisions—exactly reflected by the figure's captions "Miss the subtle target / Miss one of the target" versus "Align to both targets." Moreover, after evidence is recovered, *Refocusing* further calibrates the context window to an evidence-centric view, helping retain only the necessary surrounding context for reasoning.

D.2. Example Model Outputs

To illustrate the advantages of our approach, we present side-by-side qualitative examples comparing **DeepScan** with strong LVLMs—GPT-5 (run in its advanced reasoning mode), GPT-4o, and Qwen3-VL-235B-A22B—as well as training-free baselines *ZoomRefine* and *DyFo*. For a fair comparison across training-free methods, ZoomRefine, DyFo, and **DeepScan** all use Qwen2.5-VL-72B as the LVLM backbone. Each example reports both the visual grounding and the model's answer. The following examples confirm that **DeepScan** delivers stronger visually grounded reasoning—superior evidence localization and more accurate answers—than the two representative training-free baselines. Moreover, **DeepScan** outperforms GPT-4o and the larger-scale Qwen3-VL-235B-A22B, despite using a smaller Qwen2.5-VL-72B backbone, in both perception and reasoning. Notably, its end-to-end inference latency is of the same order as GPT-5, indicating a favorable performance–efficiency trade-off.

Example 1

Images

0.004%

Evidence

GT

(A) white

Question: What is the color of the man's cap?
(A) white (B) blue (C) red (D) black

ChatGPT 4o

What is the color of the man's cap? (A) white (B) blue (C) red (D) black

There is **no man visible in the image**, so the question *"What is the color of the man's cap?"* cannot be answered based on the provided photo. If you meant to upload a different image or if there's a specific region you'd like me to zoom in on or analyze further, please let me know!

ChatGPT 5 Thinking

What is the color of the man's cap? (A) white (B) blue (C) red (D) black

已思考 1m 27s

加载并分析图片

```
from PIL import Image, ImageOps, ImageFilter, ImageDraw, ImageStat,
img_path='/mnt/data/sa_30411.jpg'
img=Image.open(img_path)
img.size

(2048, 1360)
```

显示图片进行检查

裁剪图像左侧区域

完成

(A) white

Qwen3-VL-235B-A22B

What is the color of the man's cap? (A) white (B) blue (C) red (D) black

Qwen3-VL-235B-A22B

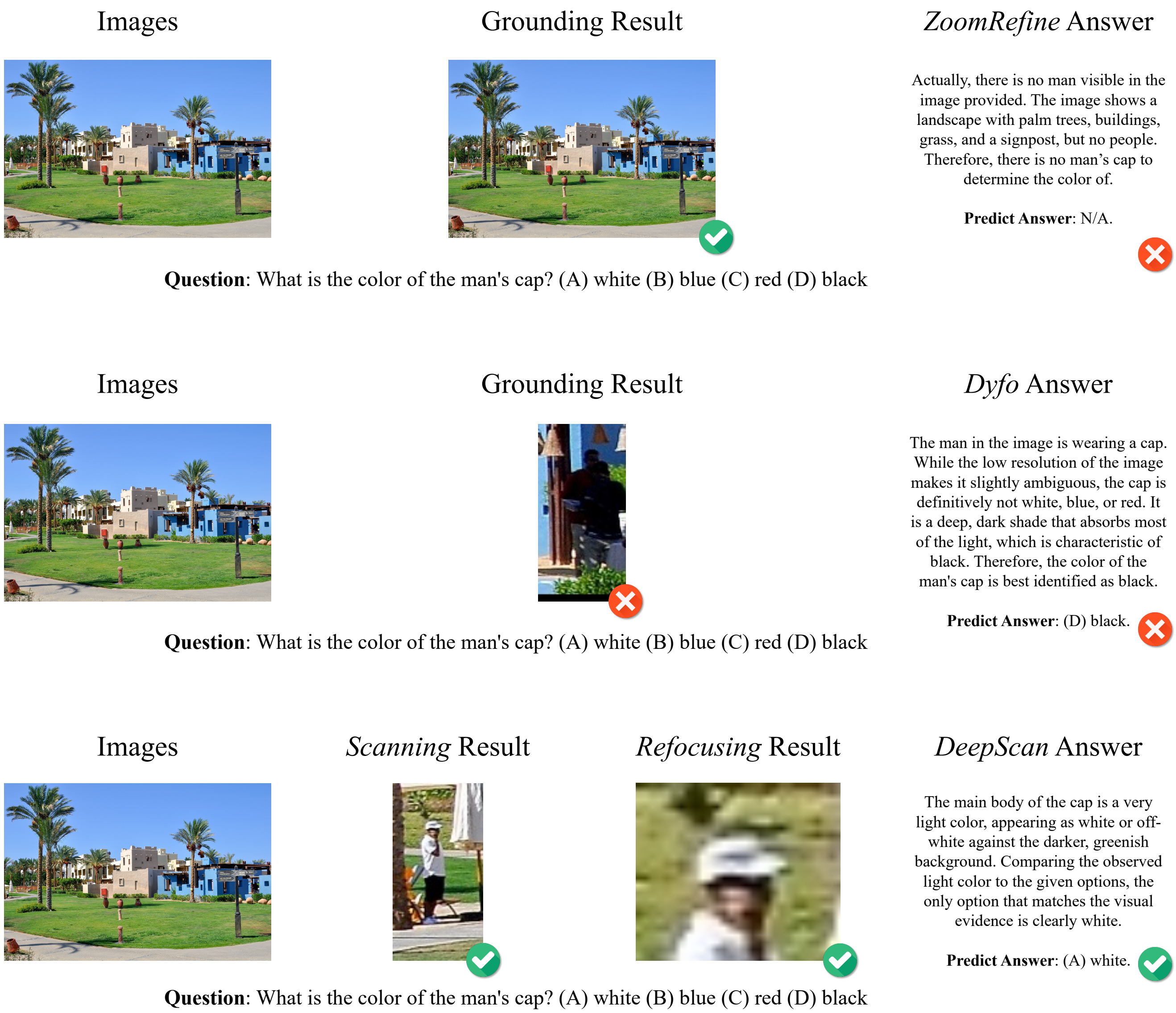
Actually, **there is no man visible in the image**. The photo shows a landscaped outdoor area with palm trees, buildings (some beige, some blue), grass, and decorative pots — but **no people**, and therefore **no man wearing a cap**.

Since the question asks about "the man's cap" but there is no man in the picture, **none of the options (A) white, (B) blue, (C) red, or (D) black can be correct**.

✅ **Correct Answer: None — there is no man in the image.**

If this is a test question, it may be designed to catch attention to detail — always check whether the subject actually exists in the image before selecting an answer.

Final Answer: Not applicable — no man present.

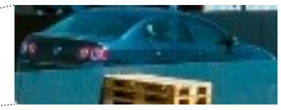

Images
Grounding Result
ZoomRefine Answer
Actually, there is no man visible in the image provided. The image shows a landscape with palm trees, buildings, grass, and a signpost, but no people. Therefore, there is no man's cap to determine the color of.
Predict Answer: N/A.
Question: What is the color of the man's cap? (A) white (B) blue (C) red (D) black
Images
Grounding Result
Dyfo Answer
The man in the image is wearing a cap. While the low resolution of the image makes it slightly ambiguous, the cap is definitively not white, blue, or red. It is a deep, dark shade that absorbs most of the light, which is characteristic of black. Therefore, the color of the man's cap is best identified as black.
Predict Answer: (D) black.
Question: What is the color of the man's cap? (A) white (B) blue (C) red (D) black
Images
Scanning Result
Refocusing Result
DeepScan Answer
The main body of the cap is a very light color, appearing as white or off-white against the darker, greenish background. Comparing the observed light color to the given options, the only option that matches the visual evidence is clearly white.
Predict Answer: (A) white.
Question: What is the color of the man's cap? (A) white (B) blue (C) red (D) black

Example 2

Images

Evidence

0.13%



GT

(B) black

Question: What is the color of the car?
(A) white (B) black (C) silver (D) red

ChatGPT 4o

What is the color of the car? (A) white (B) black (C) silver (D) red

The color of the car in the image is **silver**.

Correct answer: (C) silver.

ChatGPT 5 Thinking

What is the color of the car? (A) white (B) black (C) silver (D) red

已思考 1m 15s

加载并分析图片中的车辆颜色

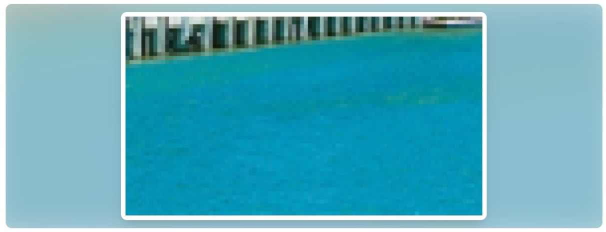
```
from PIL import Image, ImageOps, ImageFilter, ImageStat, ImageDraw,
img_path='/mnt/data/147f2f9e-e616-419f-9a27-39716652194f.png'
img=Image.open(img_path)
img.size

(872, 584)
```

显示图像并分析红色区域

缩放并显示右侧区域图像

裁剪图像区域

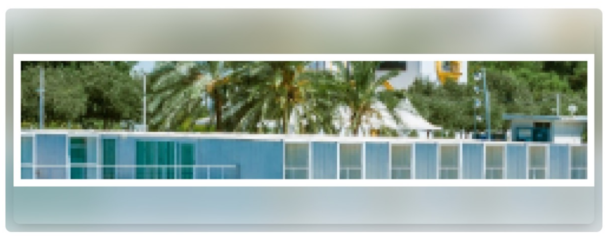
裁剪图像右侧



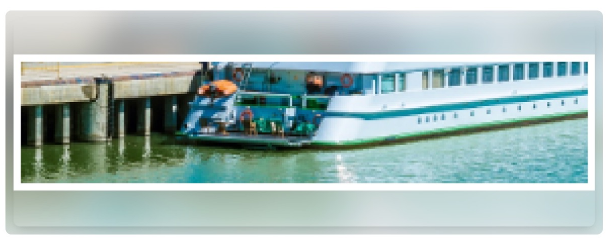
显示图像裁剪区域

查看可能的红色或蓝色车辆

裁剪图像区域



调整图像区域显示



查看左侧“蓝色汽车区域”

完成

(C) silver

Qwen3-VL-235B-A22B

What is the color of the car? (A) white (B) black (C) silver (D) red

Qwen3-VL-235B-A22B

Actually, **there is no car visible in the image**.

The image shows:

- A large river or canal with turquoise water.
- A long white and blue river cruise boat docked at a pier.
- A modern glass building on the left.
- In the background, there are trees, buildings, and a distinctive white tower with yellow trim (likely the Torre del Oro in Seville, Spain).
- The sky is clear and blue.

There are **no cars** anywhere in the scene — not on the pier, not on the road, not parked, not driving.

Therefore, **none of the options (A) white, (B) black, (C) silver, (D) red** can be correct, because **there is no car to assign a color to**.

✅ **Correct Answer: None of the above — there is no car in the image.**

(If this is a trick question or a test of observation, the intended answer may be that no car exists.)

Images

Grounding Result

ZoomRefine Answer

There is a large vehicle or structure that appears to be dark blue, which is possibly a truck, trailer, or mobile unit. Therefore, the color of the car is most likely to be black.

Predict Answer: (B) black.

Question: What is the color of the car? (A) white (B) black (C) silver (D) red

Images

Grounding Result

Dyfo Answer

Looking at the image, there is a white bus parked on the central, near the building and wooden pallets. Therefore, the color of the car appears to be white.

Predict Answer: (A) white.

Question: What is the color of the car? (A) white (B) black (C) silver (D) red

Images

Scanning Result

Refocusing Result

DeepScan Answer

The car's body panels appear very dark. The underlying paint is near-black; there's no lightness you'd expect from white or silver, and there's no saturated hue that would indicate red. The roof and trunk especially read as deep black with specular highlights, which best matches option (B).

Predict Answer: (B) black.

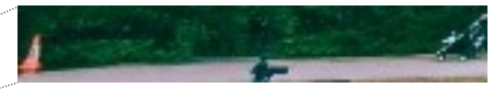
Question: What is the color of the car? (A) white (B) black (C) silver (D) red

Example 3

Images

Evidence

0.13%



GT

(B) right

Question: Is the baby carriage on the left or right side of the cone? (A) left (B) right

ChatGPT 4o

Is the baby carriage on the left or right side of the cone? (A) left (B) right

The baby carriage is on the **left** side of the orange cone.

Answer: (A) left.

ChatGPT 5 Thinking

Is the baby carriage on the left or right side of the cone? (A) left (B) right

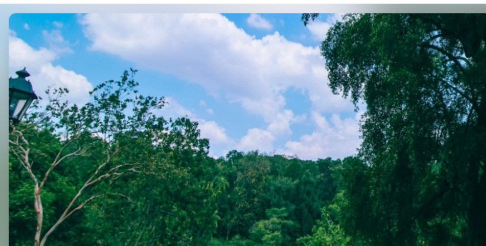
已思考 39s

打开和分析图像

```
from PIL import Image, ImageOps, ImageFilter, ImageStat, ImageDraw,
import matplotlib.pyplot as plt
import numpy as np, os, textwrap, sys, io, PIL
path="/mnt/data/sa_24971.jpg"
img=Image.open(path)
img.size

(2048, 1367)
```

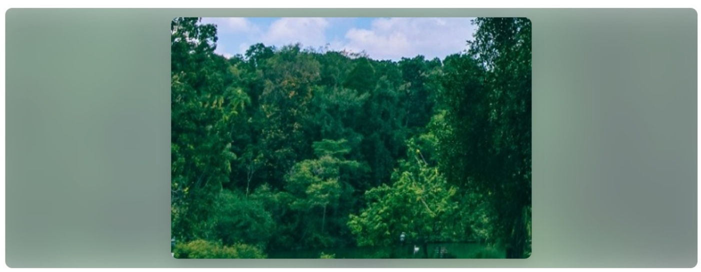
加载并分析图像

裁剪并放大左中区域

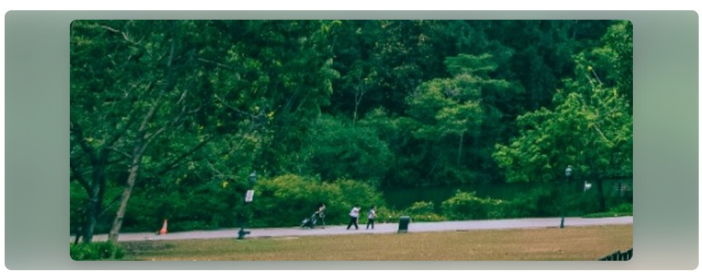


裁剪图像下左角区域

裁剪并缩放包含人物区域



调整图像裁剪区域



完成

(B) right

Qwen3-VL-235B-A22B

Is the baby carriage on the left or right side of the cone? (A) left (B) right

Qwen3-VL-235B-A22B

Actually, there is **no baby carriage** visible in the image.

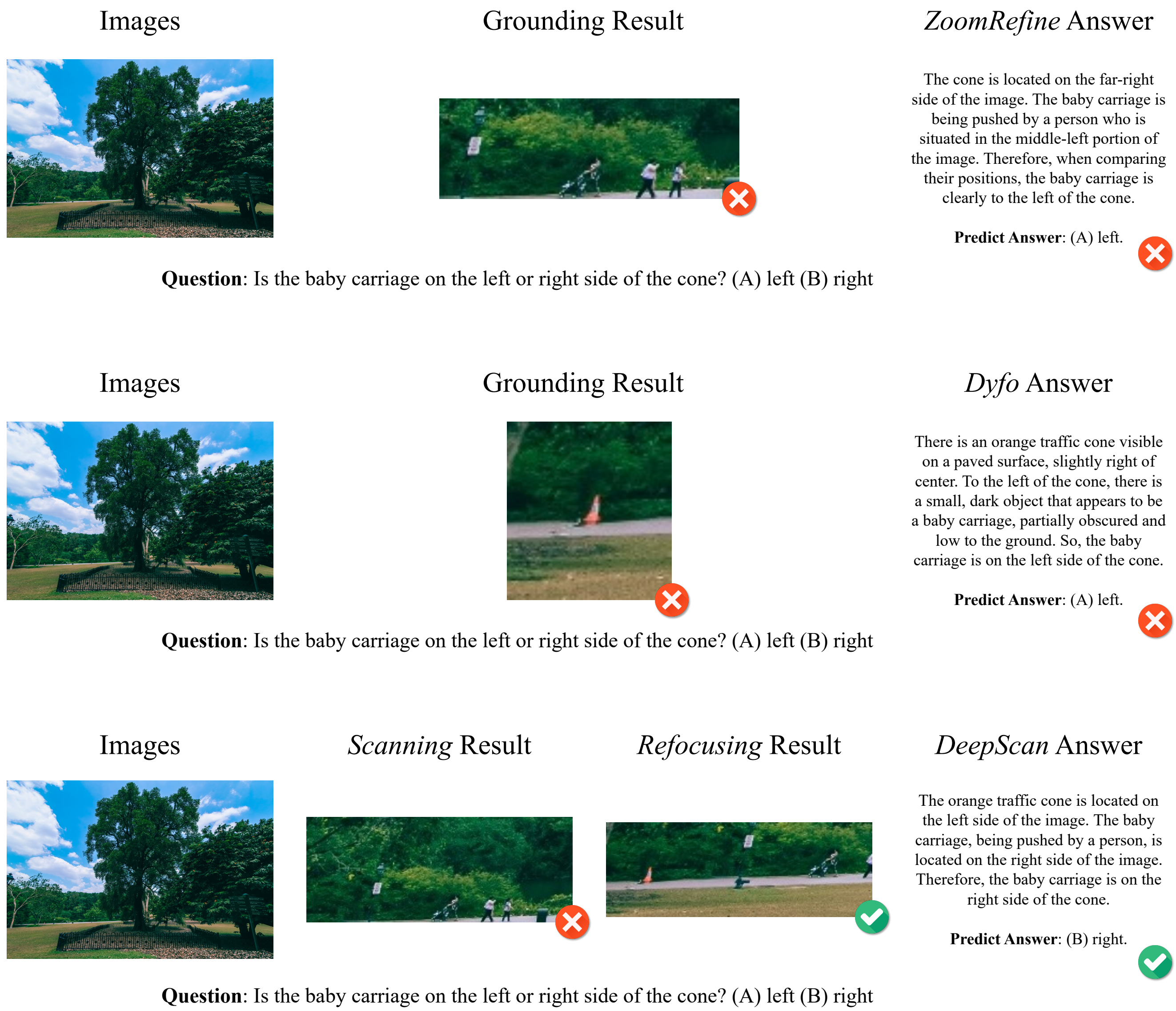
Additionally, there is **no cone** visible either.

The image shows a park scene with large trees, a black metal fence enclosing a tree base, grass, and a directional sign on the right. In the far background, there are some small figures that might be people walking, but nothing identifiable as a baby carriage or a traffic cone.

Therefore, the question cannot be answered as posed — neither object is present.

✅ Final Answer: **Neither — the baby carriage and cone are not visible in the image.**

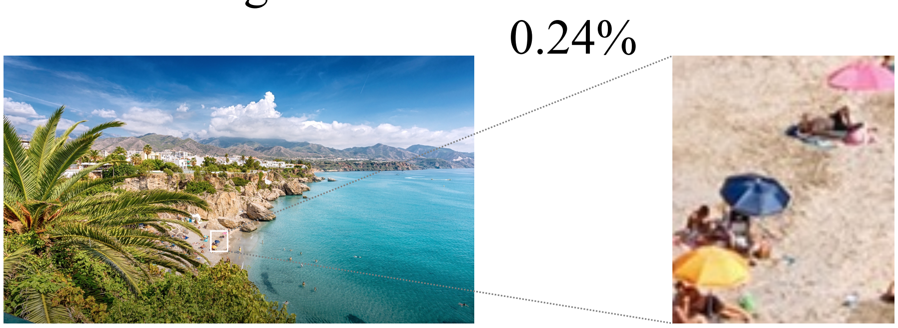
(If forced to choose between A or B without context, the correct logical response is that the premise is invalid.)


Images
Grounding Result
ZoomRefine Answer
The cone is located on the far-right side of the image. The baby carriage is being pushed by a person who is situated in the middle-left portion of the image. Therefore, when comparing their positions, the baby carriage is clearly to the left of the cone.
Predict Answer: (A) left.
Question: Is the baby carriage on the left or right side of the cone? (A) left (B) right
Images
Grounding Result
Dyfo Answer
There is an orange traffic cone visible on a paved surface, slightly right of center. To the left of the cone, there is a small, dark object that appears to be a baby carriage, partially obscured and low to the ground. So, the baby carriage is on the left side of the cone.
Predict Answer: (A) left.
Question: Is the baby carriage on the left or right side of the cone? (A) left (B) right
Images
Scanning Result
Refocusing Result
DeepScan Answer
The orange traffic cone is located on the left side of the image. The baby carriage, being pushed by a person, is located on the right side of the image. Therefore, the baby carriage is on the right side of the cone.
Predict Answer: (B) right.
Question: Is the baby carriage on the left or right side of the cone? (A) left (B) right

Example 4

Images

Evidence



GT

(B) left

Question: Is the yellow umbrella on the left or right side of the pink umbrella?
(A) right (B) left

ChatGPT 4o

Is the yellow umbrella on the left or right side of the pink umbrella?
(A) right (B) left

The yellow umbrella is on the **right** side of the pink umbrella.

Correct answer: (A) right.

ChatGPT 5 Thinking

Is the yellow umbrella on the left or right side of the pink umbrella? (A) right (B) left

已思考 1m 29s

加载和显示图像进行分析

```
from PIL import Image, ImageOps, ImageFilter, ImageStat, ImageDraw
import numpy as np, io, os, sys, PIL
img_path="/mnt/data/sa_40555.jpg"
img=Image.open(img_path)
img.size

(2048, 1162)
```

显示缩小版图像分析

裁剪并放大海滩伞区域

识别黄色和粉色雨伞

- 我看到这张图里有两把伞，一把黄色伞在左侧稍下方，另一把粉色伞在右侧靠近岩石。还有一把小粉伞在中间右边，加上一个人的身影。总体来说，黄色伞在粉色伞的左边，所以答案是“左边”。

完成

(B) left

Qwen3-VL-235B-A22B

Is the yellow umbrella on the left or right side of the pink umbrella? (A) right (B) left

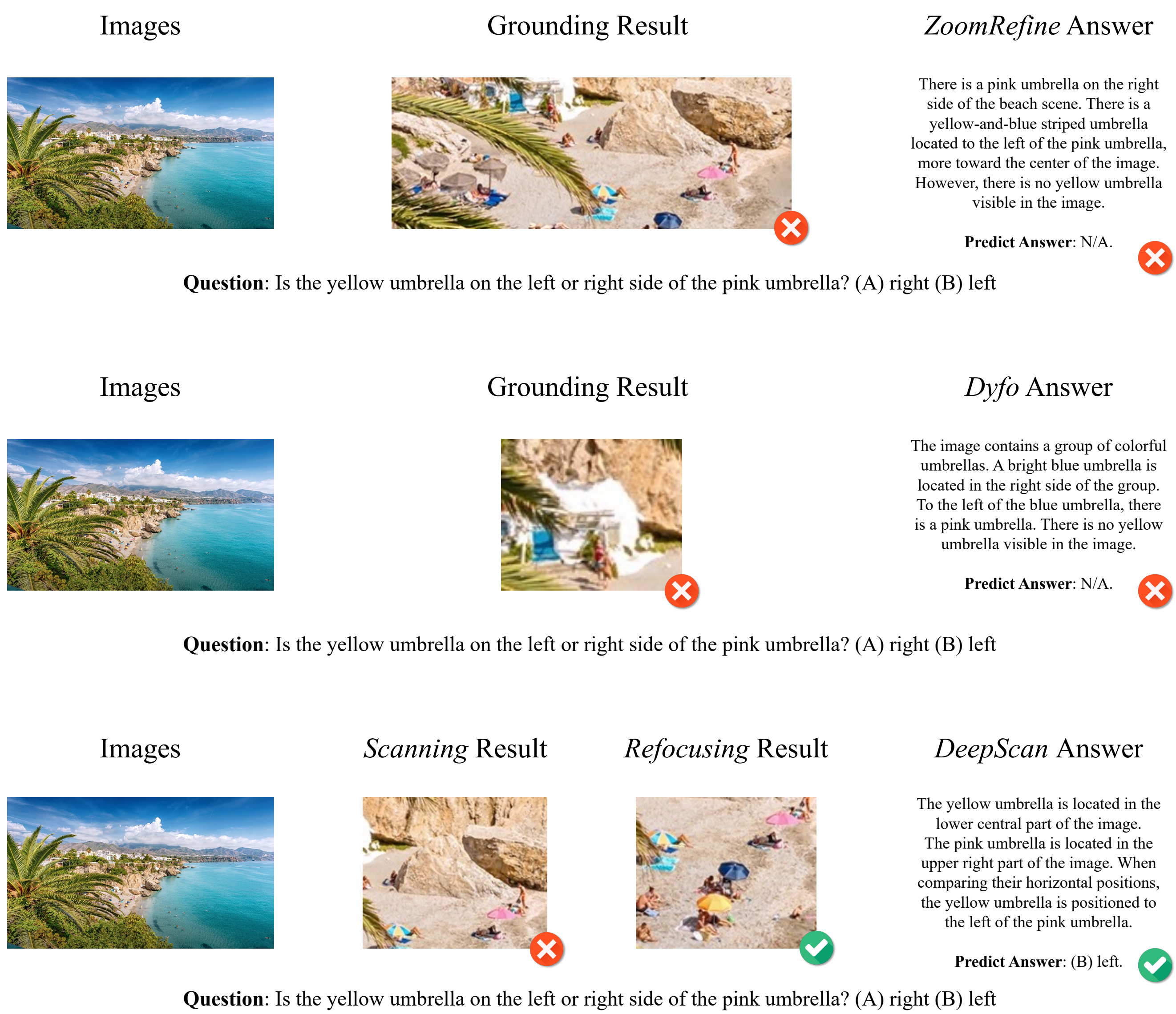
Qwen3-VL-235B-A22B

Looking at the image, we can see a beach scene with several colorful umbrellas on the sand near the water's edge.

- There is a **pink umbrella** visible among the beachgoers.
- To the **right** of that pink umbrella (from the viewer's perspective), there is a **yellow umbrella**.

Therefore, the yellow umbrella is on the **right side** of the pink umbrella.

✅ Correct answer: **(A) right**


Images
Grounding Result
ZoomRefine Answer
There is a pink umbrella on the right side of the beach scene. There is a yellow-and-blue striped umbrella located to the left of the pink umbrella, more toward the center of the image. However, there is no yellow umbrella visible in the image.
Predict Answer: N/A.
Question: Is the yellow umbrella on the left or right side of the pink umbrella? (A) right (B) left
Images
Grounding Result
Dyfo Answer
The image contains a group of colorful umbrellas. A bright blue umbrella is located in the right side of the group. To the left of the blue umbrella, there is a pink umbrella. There is no yellow umbrella visible in the image.
Predict Answer: N/A.
Question: Is the yellow umbrella on the left or right side of the pink umbrella? (A) right (B) left
Images
Scanning Result
Refocusing Result
DeepScan Answer
The yellow umbrella is located in the lower central part of the image. The pink umbrella is located in the upper right part of the image. When comparing their horizontal positions, the yellow umbrella is positioned to the left of the pink umbrella.
Predict Answer: (B) left.
Question: Is the yellow umbrella on the left or right side of the pink umbrella? (A) right (B) left