

OddGridBench: Exposing the Lack of Fine-Grained Visual Discrepancy Sensitivity in Multimodal Large Language Models

Supplementary Material

A. Appendix Outline

In these supplementary materials, we provide:

- A detailed description of the evaluation setup and full experimental results for all OddGridBench sub-tasks (Appendix B);
- Additional cross-dataset and cross-format generalization experiments are presented in Appendix C.
- Additional visualizations and qualitative examples (Appendix D).

B. Experiment Details

B.1. Model Access

This section summarizes the model access settings and parameter configurations used in our experiments (see Table 4). All results reported in this paper are based on model outputs obtained prior to November 1, 2025.

Table 4. List of MLLMs evaluated in this study, with model names shown exactly as they appear on Hugging Face or in official APIs.

Phi-3.5-vision	microsoft/Phi-3.5-vision-instruct
SAIL-VL2-2B	BytedanceDouyinContent/SAIL-VL2-2B
SAIL-VL2-8B	BytedanceDouyinContent/SAIL-VL2-8B
LLaVA-OneVision-1.5-4B	lmms-lab/LLaVA-OneVision-1.5-4B-Instruct
LLaVA-OneVision-1.5-8B	lmms-lab/LLaVA-OneVision-1.5-8B-Instruct
LLaVA-v1.6-34B	liuhaotian/llava-v1.6-34b
InternVL3.5-38B	OpenGVLab/InternVL3.5-38B
Molmo-72B	allenai/Molmo-72B-0924
Qwen2.5-VL-7B	Qwen/Qwen2.5-VL-7B-Instruct
Qwen2.5-VL-72B	Qwen/Qwen2.5-VL-72B-Instruct
Qwen3-VL-2B	Qwen/Qwen3-VL-2B-Instruct
Qwen3-VL-4B	Qwen/Qwen3-VL-4B-Instruct
Qwen3-VL-8B	Qwen/Qwen3-VL-8B-Instruct
Qwen3-VL-30B (Moe)	Qwen/Qwen3-VL-30B-A3B-Instruct
Qwen3-VL-32B	Qwen/Qwen3-VL-32B-Instruct
Gemini-2.0-Flash	gemini-2.0-flash
Gemini-2.5-Flash	gemini-2.5-flash
Gemini-2.5-Pro	gemini-2.5-pro
GPT-5	gpt-5-low

B.2. Evaluation Details

Evaluation Prompt. We use a fixed prompt that instructs the model to identify the odd element in the grid and return its position in a standardized format. We set `max_new_tokens=1024` and use default values for all other parameters.

Evaluation Prompt

You are solving an odd-one-out visual perception task.

You are given an image showing a $\{rows\} \times \{cols\}$ grid of $\{shape\}$ s. All $\{shape\}$ s appear the same, except one that is visually different in $\{odd_desc\}$.

This is a **visual perception** task that does not require lengthy logical reasoning.

Instructions

- Carefully inspect the grid.
- Identify the grid position (row and column) of the $\{shape\}$ that is different.
- Counting starts from the top-left corner, i.e., Row 1, Column 1.
- Provide brief visual observations if needed (no more than 300 words).

Output Format Requirements

- Provide concise natural-language observations.
- End the response with the final answer in the following strict LaTeX format:

```
\boxed{Row X, Column Y}
```

where X and Y are integers (e.g., Row 2, Column 3).

- Do **not** include any text after the final `\boxed{\}`.
- If no odd $\{shape\}$ exists, output:

```
\boxed{Row 0, Column 0}
```

Human Evaluation Protocol. To estimate human-level performance on OddGridBench, we conduct a human evaluation on a subset of the dataset by randomly sampling 350 instances (50 from each perceptual attribute category). Three human annotators with backgrounds in computer science and visual perception participate in the evaluation, all with normal or corrected-to-normal vision. The evaluation is conducted using a custom annotation interface implemented in Python. Images are displayed on a 4K-resolution monitor and rendered at a fixed resolution, with zooming or resizing disabled to ensure consistent perceptual conditions. Before the formal evaluation, annotators are provided with several practice examples to familiarize themselves with the task. For each image, participants are asked to identify the single grid element that differs from the others using only visual cues, without any external tools. During the eval-

uation, images are presented one at a time, and annotators select the position of the discrepant element, which may differ in color, size, rotation, or position. No feedback about correctness is provided during the evaluation.

B.3. Training Details

Training Settings. We adopt Qwen3-VL-2B-Instruct as the base model and train it using the EasyR1 reinforcement learning framework. Training is conducted with a global actor batch size of 256, micro-batch sizes of 1 (for updates) and 2 (for experience logging), and dynamic padding enabled to improve efficiency. Gradient checkpointing is enabled, and the vision tower remains fully trainable. For memory management, we use partial FSDP with rank-0 initialization enabled and no CPU offloading. During rollout, we generate $n=3$ samples per prompt using vLLM with temperature 1.0, $\text{top-}p=1.0$, and tensor parallel size 2. Validation rollouts use a lower temperature (0.6) and narrower sampling ($\text{top-}p=0.95$). The reference model is kept frozen and trained under FSDP-full-shard mode. The reward is computed in batch mode using a task-specific scoring function. Training runs for 100 optimization steps on a single node with 4 A800 GPUs. We evaluate every 10 steps, log up to 3 generated samples per validation round, and save checkpoints every 10 steps with a maximum retention of 4. For the GSPO variant, we additionally enable token-level GSPO loss, sequence-level advantage averaging, a tighter clipping ratio ($3e-4$ to $4e-4$), and disable KL regularization, following the GSPO formulation.

Training Prompt. We employ a minimal instruction prompt during training to prevent models from exploiting unnecessary reasoning cues and to ensure that learning focuses purely on visual discrepancy detection.

Training Prompt

Identify the object that differs from the others in the $\{\text{rows}\} \times \{\text{cols}\}$ grid. The difference lies in $\{\text{odd_desc}\}$. Count positions from the top-left corner as Row 1, Column 1. Return the final answer strictly in the following format:

```
\boxed{Row X, Column Y}
```

B.4. Additional Ablation Results

Impact of Grid Size on Visual Discrimination. To investigate how spatial density affects model perception, we compare MLLM accuracy across three grid configurations: small, medium, and large grids. As shown in Figure 8 and Table 5, all models experience a consistent decline in accuracy as the grid size increases. Larger grids introduce more distractors and visual clutter, which reduces the

relative salience of the odd item and makes fine-grained discrimination substantially harder. Among the evaluated models, Qwen3-VL-4B and Gemini 2.5 Pro maintain moderate robustness under mid-scale grids, while GPT-5 and InternVL 3.5-38B show steeper performance degradation. This pattern indicates that current MLLMs have limited ability to maintain spatial attention and object separation when visual scenes become more crowded.

Table 5. Performance across different grid sizes.

Model	Grid-Small	Grid-Mid	Grid-Large
InternVL3.5-38B	27.74	26.94	25.42
Qwen3-VL-32B	70.49	66.97	64.41
Gemini-2.5-Pro	54.06	47.34	41.24
GPT-5	35.51	26.79	15.82

Impact of Image Resolution on Visual Discrimination.

We further conducted an ablation study on different per-cell resolutions. As shown in Table 6, Gemini-2.5-Pro: accuracy increases from 47.0 at resolution 50 to 57.0 at 100, and then plateaus at 57.0 at 150, while GPT-5 rises from 31.0 to 35.0 to 39.0. Overall, higher resolution yields only modest gains.

Table 6. Performance across different image resolutions.

Model	Resolution		
	50	100	150
Qwen3-VL-32B	69.00	70.00	73.00
Qwen3-VL-8B	56.00	60.00	59.00
Gemini-2.5-Pro	47.00	57.00	57.00
GPT-5	31.00	35.00	39.00

Sensitivity to Perceptual Discrepancy Magnitude. Tables 12–15 summarize model performance across increasing levels of perceptual discrepancy, illustrating how accuracy changes as the visual difference becomes more pronounced.

Analysis of Localization Accuracy. Figure 11 illustrates the grid images annotated with row and column indices that are used to evaluate models’ localization performance (LabeledAcc).

Impact of Image Resolution. In a resolution ablation (50/100/150), Gemini-2.5-Pro improves 47.0→57.0 and then saturates, while GPT-5 increases 31.0→35.0→39.0, indicating only modest gains from higher resolution, as shown in Table 7.

Why Row–Column Instead of Bounding-Box Grounding? Although bounding-box localization appears to be a more fine-grained grounding objective, we intentionally

Table 7. Ablation on image resolution. We evaluate the performance of different models under three input resolutions (50, 100, and 150). Higher resolutions generally provide richer visual details and lead to improved odd-item detection accuracy.

Model	Resolution		
	50	100	150
Gemini-2.5-Pro	47.00	57.00	57.00
GPT-5	31.00	35.00	39.00

adopt the Row X, Column Y answer format in OddGridBench for two reasons. First, the OddGrid setting provides a discretized grid structure in which the notion of “location” is inherently symbolic rather than continuous; requiring pixel-level bounding boxes would not introduce additional reasoning difficulty, but rather introduce annotation ambiguity (e.g., bounding box tightness, padding, icon margins) that is irrelevant to the core perceptual challenge. Second, as shown in Table 8, even strong MLLMs that correctly predict the grid position fail almost completely when evaluated with IoU-based box matching, confirming that low-level spatial grounding remains a confounding variable that masks the actual perceptual reasoning ability we aim to measure. For these reasons, we adopt the row–column formulation to ensure that OddGridBench faithfully evaluates fine-grained visual discrimination rather than geometric box regression.

Error Analysis. We treat cases where the predicted location is close to the ground truth as a proxy for the linguistic mapping challenge (23.0% for Qwen3-VL-32B; 20.9% for Gemini-2.5-Pro). In contrast, large-deviation failures dominate (77.0% and 79.1%), suggesting the main bottleneck is visual inability.

Bounding Box Prompt

Several small grids are shown in the image, but you should focus on the global image coordinates. First, briefly describe what you observe in the image and how one object differs from the others. Then, identify the object that differs from all others in the entire image. Return its bounding box in the **global image coordinate system** strictly in the following format:

```
\boxed{[x_min, y_min, x_max, y_max]}
```

Table 8. Bounding-box localization accuracy of Qwen3-VL models at different IoU thresholds.

Model	IoU@0.3	IoU@0.4	IoU@0.5
Qwen3-VL-2B	0.074	0.053	0.036
Qwen3-VL-4B	0.089	0.062	0.041
Qwen3-VL-8B	0.102	0.078	0.055
Qwen3-VL-32B	0.133	0.087	0.053

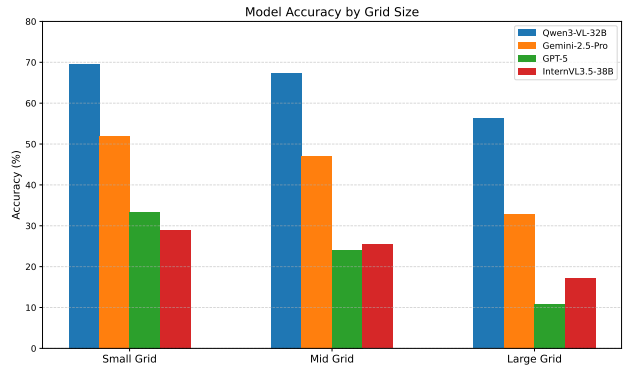


Figure 8. Comparison of MLLMs’ performance under different grid densities. Grid size is categorized by total cell count ($r \times c$): Small Grid (≤ 44 cells), Medium Grid (45–64 cells), and Large Grid (65–81 cells).

C. Additional Cross-Dataset and Cross-Format Experiments

C.1. Cross-Dataset Generalization

We conducted extended experiments on four test sets. (1) **MVTec-AD and VisA (real-image industrial anomaly detection)**. We use 13 categories from MVTec-AD and 7 from VisA, where defect images (per official ground truth) are treated as odd items and corresponding normal images as distractors. We crop background regions, keep the longest side of each cell ≥ 500 px, and organize samples into 3×3 to 5×5 grids. (2) **MNIST and Similar Chinese Characters (SCC)**. For MNIST and CC, we first select structurally similar instances as base cells (e.g., same digit with cosine similarity $\geq 80\%$ for MNIST; visually similar pairs for SCC), as shown in Figure 9. The remaining data construction procedure follows the same protocol as OddGridBench. Each dataset contains 100 grid samples, resulting in a total of 400 evaluation instances across four datasets.

We evaluate the checkpoints on these test sets, as shown in Table 9. Our method consistently outperforms both the baseline and standard GRPO across all datasets, indicating transfer beyond synthetic icons to real-image settings. We will include these extensions and further analysis in the revised version.

Table 9. Cross-dataset generalization results on four external datasets, including two real-image anomaly detection benchmarks (MVTec-AD and VisA) and two handwritten/character datasets (MNIST and SCC).

Method	MVTec-AD	VisA	MNIST	SCC
Qwen3-VL-32B	67.00	66.00	39.00	87.00
Qwen3-VL-2B (Baseline)	20.00	9.00	6.00	17.00
GRPO	47.00	39.00	18.00	54.00
OddGrid-GRPO (Ours)	49.00	40.00	37.00	60.00

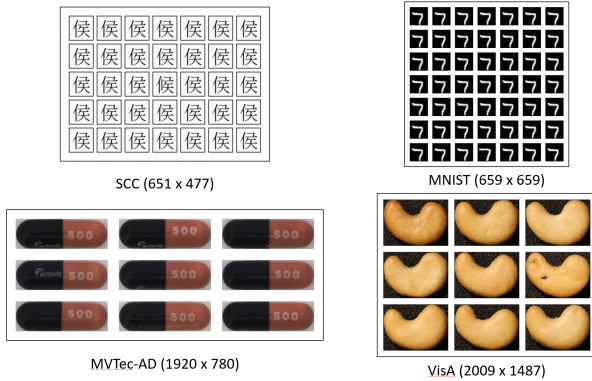


Figure 9. Visualization of the four datasets used in the image sequence settings.

Table 10. Cross-format generalization results under a non-grid setting where the model receives 8–15 independent images and predicts the odd-one-out index.

Method	MVTec-AD	VisA	MNIST	SCC
Qwen3-VL-32B	76.00	72.00	62.00	86.00
Qwen3-VL-2B (Baseline)	29.00	32.00	2.00	28.00
GRPO	42.00	38.00	11.00	51.00
OddGrid-GRPO (Ours)	46.00	40.00	16.00	64.00

C.2. Cross-Format Generalization

We evaluate a non-grid variant (as shown in Figure 10) where the model receives 8–15 separate images and predicts the odd-one-out index, Table 10. OddGrid-GRPO outperforms both the baseline and standard GRPO, including on real-image anomaly detection datasets (MVTec-AD/VisA), suggesting gains beyond grid-specific memorization. We will add qualitative examples in the revised version.

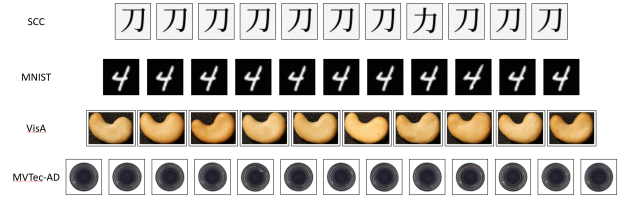


Figure 10. Visualization of the four datasets used in the row and column settings.

Evaluation Prompt (Row and Column type)

An image containing a grid of objects is provided. Identify the object that differs from the others in the $rows \times cols$ grid. Cells are indexed starting from the top-left corner as **Row 1, Column 1**. Return your answer strictly in the following format:

`\boxed{Row X, Column Y}`

Image Sequence Identification Prompt

You are presented with N images, labeled $image_1, image_2, \dots, image_N$. Identify all anomalous images in the set.

Output Rules:

1. You may perform observation and comparative reasoning before answering.
2. The final answer must be enclosed in exactly one `\boxed{}` block.
3. Inside the box, list the labels of all anomalous images (e.g., $image_1, image_3$).
4. If no anomalous images are found, output `\boxed{}`.

C.3. Evaluation on Real-World Anomaly Detection Benchmarks.

We argue that discrepancy perception is attribute-agnostic, and current models do not yet fully possess this ability. To better reflect real-world industrial inspection, we redesign the anomaly-count distribution on MVTec-AD and VisA by additionally including all-normal cases and cases with two anomalous items, as shown in Table 11. We also revise the prompt to require the model to output the locations of all anomalies. Under EM/F1, even Qwen3-VL-32B reaches only 47.08/58.46 on MVTec-AD and 40.00/53.48 on VisA, indicating that robust real-image discrepancy detection remains challenging.

Table 11. Evaluation on real-world industrial anomaly detection datasets. We report Exact Match (EM) and F1 scores on MVTec-AD and VisA.

Model	MVTec-AD		VisA	
	EM	F1	EM	F1
Qwen3-VL-8B	37.18	48.63	29.71	43.30
Qwen3-VL-32B	47.08	58.46	40.00	53.48

D. Example Data and Model Outputs

Figures 12 to 17 show examples from OddGridBench and the responses of Qwen3-VL-32B and Gemini-2.5-Pro. Figures 18 to 20 show examples from OddGridBench and the responses of Qwen3-VL-2B and the oddGrid-GRPO trained model.

Table 12. Accuracy across increasing discrepancy levels on different odd types (Qwen3-VL-32B) Higher Δ levels correspond to larger visual differences.

Odd Type	Δ Level 1	Δ Level 2	Δ Level 3	Δ Level 4	Δ Level 5
Color	53.66	79.07	92.31	97.96	96.55
Position	31.25	28.57	44.26	53.85	63.64
Rotation	38.10	31.48	67.65	70.59	69.44
Size	7.50	25.00	50.00	52.17	67.86

Table 13. Accuracy across increasing discrepancy levels on different odd types (Gemini-2.5-Pro). Higher Δ levels correspond to larger visual differences.

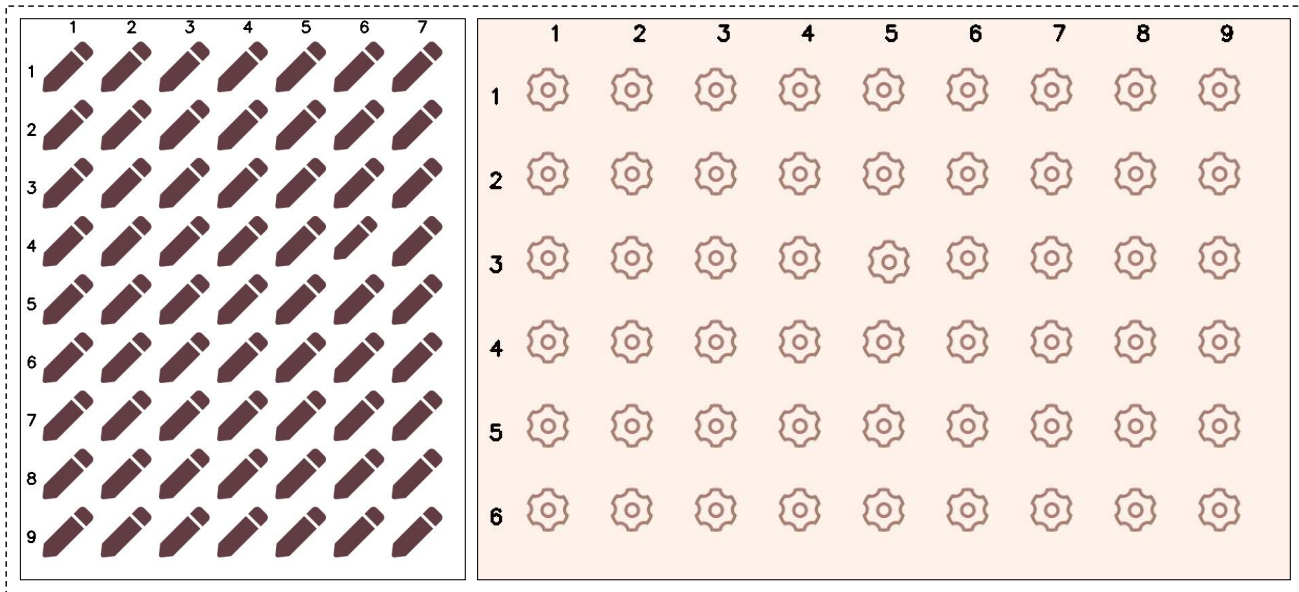
Odd Type	Δ Level 1	Δ Level 2	Δ Level 3	Δ Level 4	Δ Level 5
Color	58.54	69.77	86.54	89.80	94.83
Position	6.25	4.29	9.84	3.85	9.09
Rotation	14.29	16.67	32.35	41.18	33.33
Size	2.50	5.00	10.87	13.04	17.86

Table 14. Accuracy across increasing discrepancy levels on different odd types (GPT-5). Higher Δ levels correspond to larger visual differences.

Odd Type	Δ Level 1	Δ Level 2	Δ Level 3	Δ Level 4	Δ Level 5
Color	39.02	53.49	59.62	57.14	67.24
Position	0.00	2.86	8.20	7.69	9.09
Rotation	4.76	22.22	17.65	26.47	36.11
Size	0.00	12.50	4.35	15.22	17.86

Table 15. Accuracy across increasing discrepancy levels on different odd types (InternVL3.5-38B). Higher Δ levels correspond to larger visual differences.

Odd Type	Δ Level 1	Δ Level 2	Δ Level 3	Δ Level 4	Δ Level 5
Color	14.63	44.19	51.92	51.02	60.34
Position	9.38	5.71	9.84	19.23	18.18
Rotation	0.00	11.11	14.71	32.35	27.78
Size	0.00	5.00	2.17	0.00	3.57



Example of Image with Number

Figure 11. Example of grid images with row and column indices used in OddGridBench.

Input image

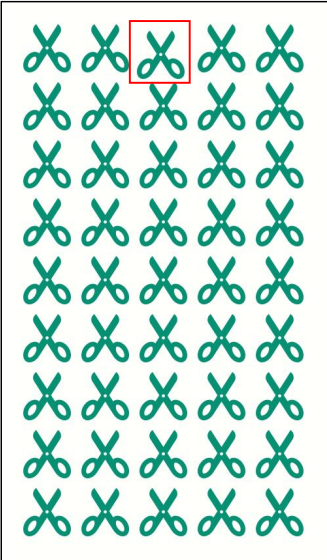
Question: Identify the grid position (row and column) of the ruler that is different.

Qwen3-VL-32B:
 Upon careful inspection of the 7x6 grid of rulers, all rulers appear identical in design, orientation, and size — each is a simple, diagonally oriented ruler icon with tick marks and a small circular hole near one end. The rulers are uniformly spaced and aligned in rows and columns. However, one ruler stands out due to a subtle difference in its **orientation**. While all other rulers are tilted in the same diagonal direction (from top-left to bottom-right), the ruler in **Row 4, Column 2** is rotated slightly differently — it appears to be tilted in the opposite diagonal direction (from top-right to bottom-left), making it visually distinct. This difference is not in shape or size, but in **angle or rotation**, which breaks the consistent pattern across the grid. All other rulers maintain the same diagonal slant, reinforcing that this one is the odd one out. No other rulers show variation in color, tick marks, or position within the grid. The grid is otherwise perfectly uniform. Thus, the visually different ruler is located at: **Row 4, Column 2** ❌

Gemini-2.5-Pro:
 The image contains a 7x6 grid of rulers. All rulers are oriented diagonally in the same direction. Upon careful inspection of the alignment and spacing of the rulers, a single ruler is found to be out of position. Most rulers are placed with uniform horizontal and vertical spacing, creating a regular pattern. However, the ruler in the fourth row and the fifth column is shifted slightly to the left relative to the other rulers in its column and row. This misalignment disrupts the otherwise consistent grid structure. All other rulers maintain their correct positions within the grid. **Row 4, Column 5** ❌

Ground Truth: Row 2, Column 3

Figure 12. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 1/6).



Input image

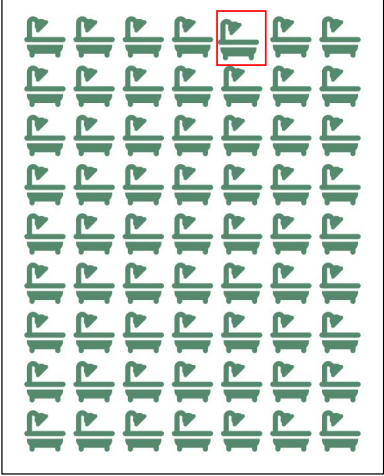
Question: Identify the grid position (row and column) of the scissors that is different.

Qwen3-VL-32B:
 Upon careful inspection of the 9×5 grid of scissors, all scissors appear identical in shape, orientation, and color — teal with a classic scissor design — except for one subtle difference in **position**. Each row contains 5 scissors, and each column contains 9 scissors. The scissors are arranged in a regular grid, but upon close examination, the scissors in **Row 3, Column 3** is slightly misaligned — it appears to be rotated or shifted compared to the others. While the rest of the scissors are oriented with the handles pointing downward and blades upward (in a standard "X" shape), this particular pair seems to have a slight rotation or positional offset, making it stand out visually. This difference is not in color or shape, but in **spatial alignment** — it does not sit perfectly in the grid cell, appearing slightly tilted or offset vertically or horizontally relative to its neighbors. This misalignment breaks the uniformity of the otherwise perfectly aligned grid. All other scissors are perfectly aligned and identical in orientation, confirming that this is the only odd one out. **Row 3, Column 3** ❌

Gemini-2.5-Pro:
 The image displays a 9x5 grid of green scissors. Upon careful inspection of the alignment of each scissors icon, a subtle anomaly can be detected. While the vast majority of the scissors are arranged in a regular, evenly spaced grid, one icon is slightly out of place. Specifically, the scissors located in the seventh row and the fourth column is shifted slightly downwards relative to the other scissors in the same row and the grid as a whole. This breaks the otherwise consistent vertical and horizontal alignment of the pattern. **Row 7, Column 4** ❌

Ground Truth: Row 1, Column 3

Figure 13. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 2/6).



Input image

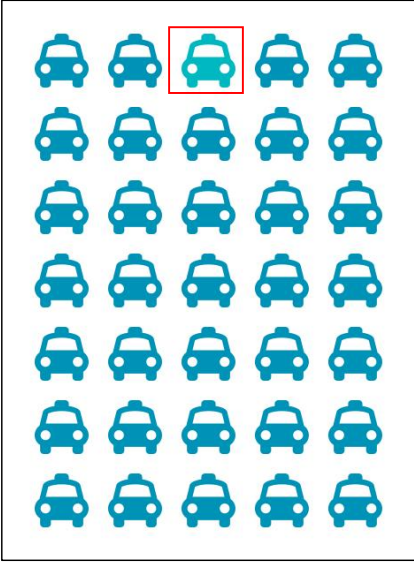
Question: Identify the grid position (row and column) of the bath that is different.

Qwen3-VL-32B:
 Upon careful inspection of the 9×7 grid of green bath icons, each bath appears identical at first glance — all have the same shape, color, and orientation, with a small heart-shaped object placed inside the tub, near the left side (from the viewer's perspective). However, one bath stands out due to a subtle difference in the position of the heart-shaped object. While in all other baths the heart is consistently placed near the left end of the tub, in one specific bath, the heart is shifted slightly to the right — closer to the center of the tub. This shift is small but noticeable upon close examination. Scanning row by row from top to bottom, and column by column from left to right, the anomaly is located in **Row 8, Column 4**. In this position, the heart is visibly more centered compared to the surrounding baths where it remains consistently left-aligned. All other visual attributes — including the outline of the bath, the legs, the faucet, and the overall color — remain identical across the grid. Therefore, the only distinguishing feature is the misaligned heart in that one bath. This is a classic visual perception task where attention to fine detail is required. The difference is minimal, but consistent across all other baths, making the outlier detectable through careful comparison. **Row 8, Column 4** ❌

Gemini-2.5-Pro:
 The image displays a 9x7 grid of identical green bath icons. Upon careful examination of the alignment and spacing of the icons, a single anomaly can be identified. One of the baths is not correctly positioned vertically within the grid structure. Specifically, the bath located in the seventh row and the fourth column is shifted slightly upward compared to all the other icons. This misalignment disrupts the consistent vertical spacing observed throughout the rest of the grid. **Row 7, Column 4** ❌

Ground Truth: Row 1, Column 5

Figure 14. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 3/6).



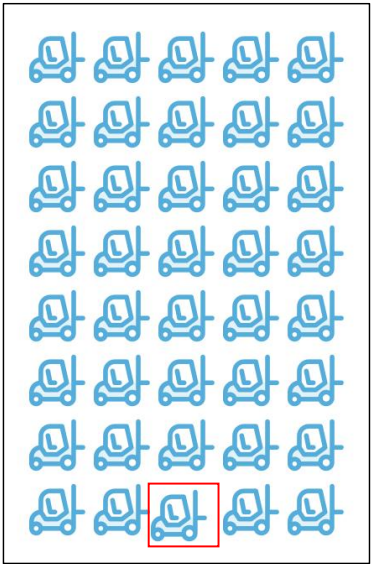
Question: Identify the grid position (row and column) of the taxi that is different.

Qwen3-VL-32B:
 Observing the 7×5 grid of taxis, all appear identical in shape and orientation. However, one taxi stands out due to a subtle difference in color. Most taxis are a uniform teal-blue, but one is slightly lighter and more cyan in hue, making it visually distinct.
 This different taxi is located in the top row (Row 1). Scanning from left to right in Row 1, the first four taxis are the same teal-blue as the rest of the grid. The fifth taxi, in Column 5, is visibly lighter and more cyan. This color variation is consistent across the entire image, with no other taxis showing a similar hue. The difference is noticeable upon close inspection but not immediately obvious at a glance.
 All other taxis in the grid match the standard teal-blue color. There are no differences in shape, size, or orientation. The anomaly is purely chromatic.
 Thus, the odd-one-out is located at Row 1, Column 5. ✗

Gemini-2.5-Pro:
 The image displays a 7×5 grid of taxi icons. All the taxis are a solid dark teal color, with one exception. The taxi located in the first row and third column is a lighter shade of teal, with a noticeable green tint and a subtle color gradient. This difference in color and shading makes it the odd one out in the grid. ✓

Input image
Ground Truth: Row 1, Column 3

Figure 15. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 4/6).



Question: Identify the grid position (row and column) of the forklift that is different.

Qwen3-VL-32B:
 Upon careful inspection of the 8×5 grid of forklifts, all forklifts appear identical in design, orientation, and color—except for one. The forklifts are uniformly arranged, each facing right with a consistent shape, including the cabin, forks, and wheels.
 The anomaly is located in the **last row (Row 8)**. Scanning from left to right, the first six forklifts in Row 8 appear normal. However, the **third forklift** in this row (Column 3) is visually different: its cabin is tilted or rotated slightly downward, and the fork structure appears misaligned or partially obscured, giving it a distorted appearance compared to the others. This distortion is subtle but noticeable upon close inspection.
 All other forklifts in the grid maintain consistent orientation and structure. The difference in the third forklift of Row 8 is the only deviation in the entire grid.
 Thus, the odd-one-out is located at: ✓ **Row 8, Column 3**

Gemini-2.5-Pro:
 The image contains a grid of 40 forklifts, arranged in 8 rows and 5 columns. Upon close inspection, nearly all the forklifts are perfectly aligned both vertically and horizontally, forming a regular pattern. However, one forklift deviates from this pattern. The forklift located in the seventh row and the second column is shifted slightly to the left of its intended position, breaking the vertical alignment with the other forklifts in that column. All other forklifts maintain their correct spacing and alignment. ✗

Input image
Ground Truth: Row 8, Column 3

Figure 16. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 5/6).

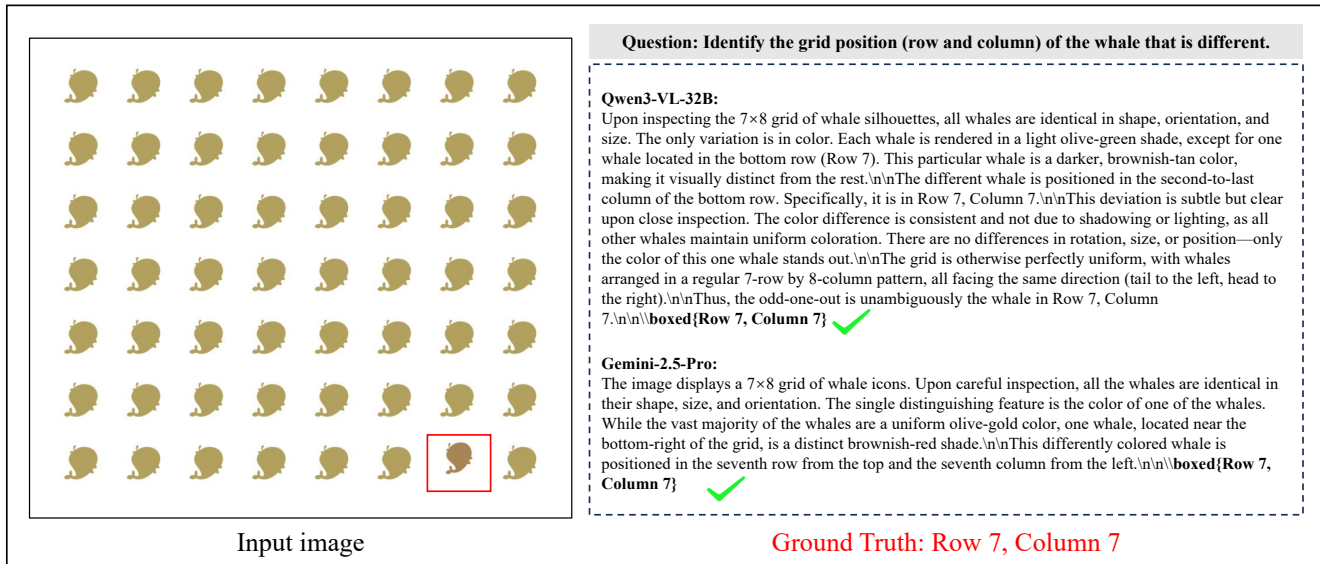


Figure 17. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-32B and Gemini-2.5 Pro (Example 6/6).

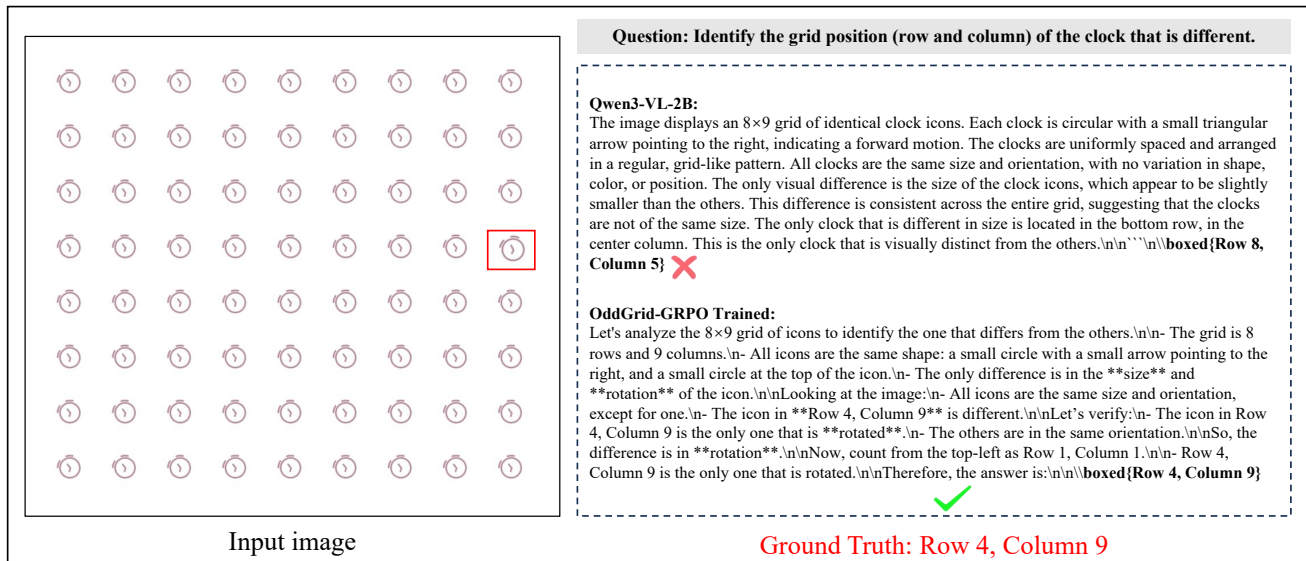


Figure 18. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-2B and oddGrid-GRPO trained model (Example 1/3).

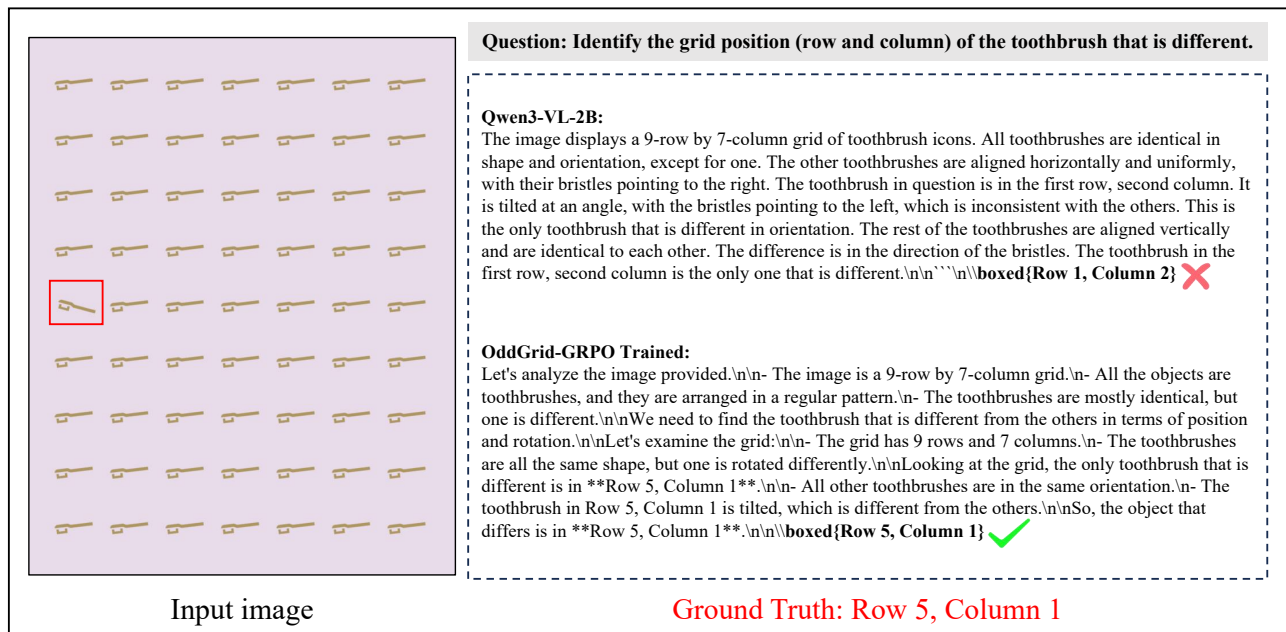


Figure 19. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-2B and oddGrid-GRPO trained model (Example 2/3).

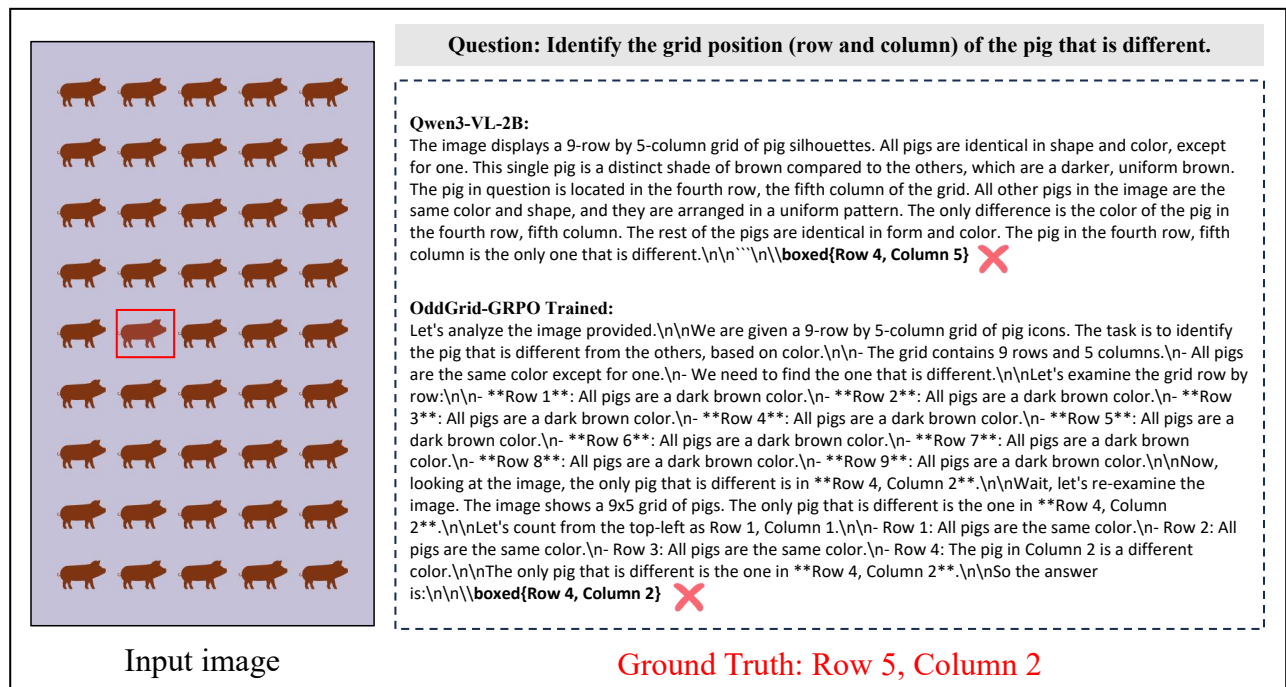


Figure 20. Example of model outputs for fine-grained perceptual discrimination on OddGridBench, showing predictions from Qwen3-VL-2B and oddGrid-GRPO trained model (Example 3/3).