ROTATED, BUT HOW FAR? DIAGNOSING AND IMPROVING OBJECT-ROTATION REASONING IN VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) can detect that an object has rotated across views, but cannot reliably tell by how much. We introduce OR-Bench, a fine-grained benchmark for object-rotation reasoning with eight tasks covering rotation detection, rotation magnitude estimation, and multi-view rotation reasoning. Across 12 VLMs, the gap is stark: the strongest models approach 100% accuracy on detection, yet even coarse magnitude estimation is near chance. When asked for exact angles, models place 91.8–100% of their predictions on just 0°, 90°, and 180°, a failure we term *canonical-angle collapse*. This collapse persists even without visual input. Representation probing shows that missing information is only part of the explanation. Although rotation information becomes less recoverable at finer granularity, substantial coarse-grained information remains, and a simple linear probe outperforms the models' generated answers. This suggests that VLMs underuse rotation information they already encode. We therefore propose RotationCue, a lightweight decoder that recovers coarse rotation information from the VLM's own frozen representations and feeds it back to the model as intermediate textual context. Across three VLMs, RotationCue improves every model–task combination on OR-Bench, raising macro-average accuracy by 7.9–12.6 points while preserving general capabilities.

1 INTRODUCTION

Vision-language models (VLMs) (Zhang et al., 2024) are moving from static image understanding toward perception and reasoning in dynamic environments (Liu et al., 2023; Chen et al., 2024; Tong et al., 2024), where a model must not only recognize what an object is, but also understand how it has changed between observations. Object rotation[1] is one such basic and common change. A rotated object can look completely different from one view to the next, yet it is the same object. This makes object rotation a natural test of spatial reasoning (Shepard & Metzler, 1971). Humans handle this readily: given two views of the same object, people can typically tell whether it has been rotated, roughly estimate by how much, and distinguish smaller rotations from larger ones. Can VLMs do the same?

Despite its importance, object rotation remains underexplored in VLM research. Recent benchmarks (Zhang et al., 2026; Kong et al., 2025) include rotation-related tasks and consistently identify rotation as a weakness of VLMs. Yet they tell us mainly that models fail, not how: rotation is typically treated as a subtask within a broader benchmark, with coarse evaluation and little analysis of where and why errors arise. We introduce the Object Rotation Benchmark (**OR-Bench**) to study object rotation systematically, with eight tasks organized into two groups (Figure 2). Five two-view tasks ask how precisely a model understands the rotation between two observations, from detecting whether it occurred to estimating its coarse angular range and exact angle. Three multi-view tasks ask how well it uses rotation relations across observations: matching views to a reference, comparing rotation magnitudes, and predicting rotated views. Together, these tasks examine how precisely VLMs estimate rotation and how effectively they use it.

Our evaluation reveals a sharp boundary (Figure 1): VLMs can tell *whether* an object has rotated, but not *how far*. Rotation Detection averages 87.8% across 12 models, with strong models approaching

[1] *Object rotation* here means an object's azimuthal rotation about its own vertical axis, with the camera fixed; it is not in-plane image rotation. Section 3 gives the formal definition.

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

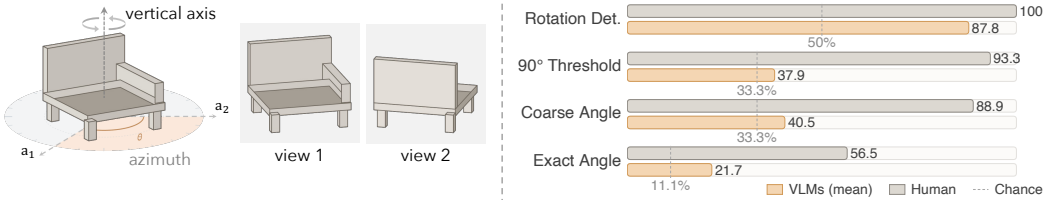


Figure 1: Object-rotation reasoning in OR-Bench. Left: two views of the same object under azimuth rotation. Right: VLMs detect rotation reliably, but performance drops sharply once rotation magnitude must be estimated, even coarsely, leaving a large gap to humans.

perfect accuracy. Performance drops sharply once models must estimate rotation magnitude, even coarsely: average accuracy falls to 37.9% on 90° Threshold (comparing the rotation angle with 90°) and 40.5% on Coarse Angle (estimating its approximate angular range), barely above the 33.3% chance level and far below human accuracy (93.3% and 88.9%, respectively). The failure becomes even more pronounced on the Exact Angle tasks (selecting or directly predicting the exact angle): across models, 91.8–100% of predictions concentrate on just three canonical angles (0°, 90°, and 180°), a failure mode we term **canonical-angle collapse**. This collapse persists even when the visual input is removed entirely.

To understand these failures, we probe the models' internal representations and identify two limitations. First, as angular granularity becomes finer, the models' sensitivity to rotation progressively declines, and less rotation information can be recovered from representations. Second, even where the information is encoded, the models do not fully exploit it: on Coarse Angle in particular, a simple linear probe on the same representations substantially outperforms the models' own predictions.

If rotation information is encoded but underused, can we put it to work? We propose **RotationCue**, which takes this information from recoverable to usable. Using external multi-view data, it trains a lightweight decoder to extract coarse-grained, structured rotation cues from a frozen VLM's visual representations. These cues are then fed back to the VLM as intermediate context to support its reasoning. Across three VLMs, RotationCue improves accuracy on all eight OR-Bench tasks, raising the macro-average (Macro-8) by 7.9–12.6 points. With RotationCue, Qwen3-VL-8B outperforms strong proprietary models and a supervised fine-tuning baseline, at modest training cost and without sacrificing general capabilities. Our contributions are threefold:

- We present OR-Bench, a fine-grained benchmark for object-rotation reasoning that reveals systematic failures in rotation estimation, exhibiting canonical-angle collapse.
- We trace these failures to two limitations through representation probing: rotation information is less recoverable at finer angular granularity, and where it is recoverable, VLMs fail to fully exploit it.
- We propose RotationCue, which recovers structured rotation information from frozen representations and reinjects it as intermediate context, improving object-rotation reasoning across tasks and VLMs.

2 Related Work

Spatial reasoning in VLMs. Spatial reasoning benchmarks for VLMs have progressed from pairwise relations in single images (Kamath et al., 2023; Tong et al., 2024) to spatial understanding (Ma et al., 2025; Wang et al., 2025b) and multi-view, video, and dynamic settings (Yang et al., 2025; Zhou et al., 2025; Jia et al., 2026), covering spatial relations such as relative position, distance, egocentric direction, and perspective taking. Most closely related, SpinBench (Zhang et al., 2026) decomposes perspective taking into fine-grained diagnostic categories, including rotation, and reveals poor rotational understanding across a wide range of VLMs. LRR-Bench (Kong et al., 2025) evaluates rotation alongside spatial movement on synthetic data and likewise identifies it as a persistent weakness. Building on these findings, OR-Bench treats object rotation as a research problem in its own right, evaluating rotation reasoning across controlled angular granularities. Beyond accuracy, we conduct in-depth error and representation analyses.

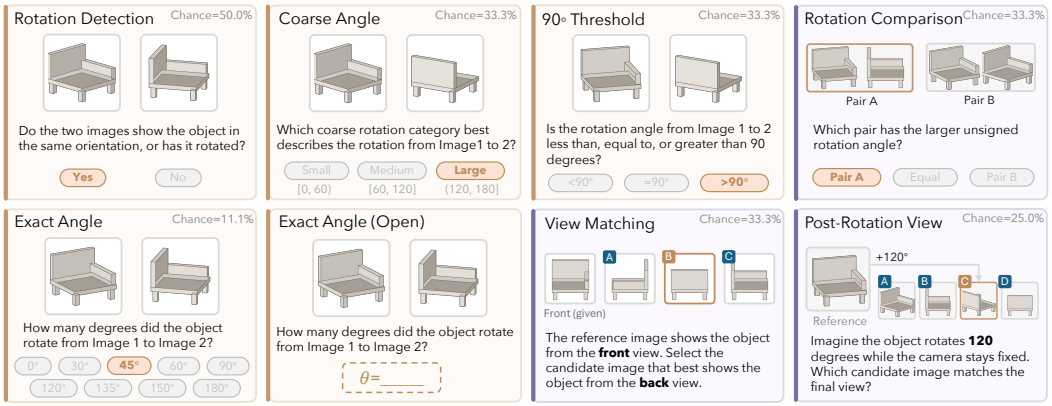


Figure 2: **Overview of OR-Bench.** Eight tasks span two-view rotation understanding (orange) and multi-view rotation use (purple), covering multiple levels of answer granularity, from detecting whether a rotation occurred to specifying its exact angle.

Probing VLM representations. Linear probing (Belinkov, 2022) is a standard tool for analyzing internal model representations. Prior work has used it to study depth and surface geometry in vision foundation models (Danier et al., 2025), as well as visual and spatial information in VLMs, such as chart coordinates, object positions, and orientations (Fu et al., 2025; Tartaglini et al., 2025; Kang et al., 2026; Theodoridis et al., 2026). We also adopt linear probing, but go beyond asking whether rotation information is present: we examine what kinds of rotation information are encoded, from coarse to fine, and how well each can be recovered. Building on this analysis, we propose RotationCue, which extracts rotation information from the VLM's own frozen representations and feeds it back as intermediate context. Unlike existing intervention methods (Liu et al., 2026; Li et al., 2023), RotationCue neither alters the visual input nor directly edits internal activations, improving rotation reasoning while preserving general-task performance.

3 OR-Bench: Evaluating Object-Level Rotation

Problem Definition. OR-Bench studies object rotation about the vertical axis (a change in azimuth), rather than an in-plane rotation of the image. Most objects rest on a supporting surface, and such rotation is the most common in everyday observation. Fixing this axis gives a controlled setting in which to evaluate the object rotation reasoning of VLMs. Given azimuths a_1 and a_2 of two views of the same object, the rotation angle is

$$\theta = \min\left(|a_1 - a_2|,\ 360^\circ - |a_1 - a_2|\right) \in [0^\circ, 180^\circ].$$

The angle is unsigned; when a task involves rotation direction, it is provided explicitly in the text.

Data Construction. OR-Bench uses the 360° turntable subset of the Amazon Berkeley Objects (ABO) (Collins et al., 2022), with 72 views per product scan at 5° intervals. Frame offsets give the exact angle, so every task carries a noise-free label. From an initial pool of 100 objects, we remove those whose orientation is visually ambiguous due to symmetry and retain 73, from which we build 2,109 test examples around nine reference angles: $\{0^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ, 120^\circ, 135^\circ, 150^\circ, 180^\circ\}$. Closed-set tasks balance target classes and answer positions; image-selection candidates use different views of the same object, so the choice depends on rotation rather than object identity.

Two-view tasks. Five tasks reason over image pairs and assess rotation understanding at different output granularities. *Rotation Detection* determines whether a rotation occurs; *Coarse Angle* classifies the rotation as small ($[0^\circ, 60^\circ)$), medium ($[60^\circ, 120^\circ]$), or large ($(120^\circ, 180^\circ]$); 90° *Threshold* distinguishes rotations below, equal to, or above 90°; and *Exact Angle (MCQ)* and *Exact Angle (Open)* require selecting or directly predicting the exact angle, respectively. Coarse Angle and the two Exact Angle variants use the same image pairs, enabling direct comparison across output granularities.

Table 1: Results (%) on OR-Bench. Exact Angle (Open) uses exact-match accuracy. **Bold** and underlined values indicate the best and second-best results within each group. Blue and red indicate above- and below-chance performance, respectively; darker shades indicate larger deviations.

Model	Rotation Detection	90° Thresh.	Coarse Angle	Exact Angle (MCQ)	Exact Angle (Open)	View Matching	Rotation Comparison	Post-Rot. View	Micro -8	Macro -8
Proprietary Multimodal Large Language Models										
GPT-5.4 (OpenAI, 2026)	97.9	40.3	**53.1**	**34.6**	**39.5**	**75.9**	**56.4**	**37.5**	**59.6**	**54.4**
GPT-4o (OpenAI, 2024a)	**99.2**	37.9	42.0	27.2	26.3	68.5	49.4	36.5	54.2	48.4
GPT-4o-mini (OpenAI, 2024b)	97.9	**42.0**	38.3	22.2	26.3	59.3	40.3	27.1	50.6	44.2
Open-source Multimodal Large Language Models										
Qwen3-VL-8B (Bai et al., 2025a)	**99.6**	37.0	40.7	**24.7**	25.5	**65.3**	**48.6**	32.8	**53.0**	**46.8**
InternVL3-8B (Zhu et al., 2025)	96.5	39.1	42.0	22.2	**28.4**	47.2	39.5	29.2	49.5	43.0
InternVL3.5-8B (Wang et al., 2025a)	99.2	37.4	32.1	22.2	23.5	49.1	44.0	**36.5**	49.5	43.0
Qwen2.5-VL-7B (Bai et al., 2025b)	86.0	37.4	41.6	22.2	23.0	56.0	41.6	30.2	47.4	42.3
InternVL3.5-4B (Wang et al., 2025a)	97.7	**41.2**	38.3	20.6	22.6	48.6	42.0	26.0	48.8	42.1
Qwen2-VL-7B (Wang et al., 2024)	62.8	38.3	**45.7**	16.0	20.2	54.6	34.2	27.6	40.4	37.4
Cosmos-Reason1-7B (Azzolini et al., 2025)	65.0	36.2	38.3	21.0	22.2	52.8	36.6	25.0	40.4	37.1
Phi-3.5-Vision (Abdin et al., 2024)	90.7	33.7	37.9	16.5	12.3	38.4	32.9	24.0	42.4	35.8
LLaVA-OneVision-7B (Li et al., 2024)	60.5	33.7	35.8	11.1	14.0	37.0	35.0	25.5	35.0	31.6
Human	100.0	93.3	88.9	56.5	56.5	100.0	72.1	87.2	–	81.8

Multi-view tasks. Three tasks use rotation relations across multiple observations. *View Matching* identifies a specified view (e.g., front, back, left, or right) from candidate images; *Rotation Comparison* compares the magnitudes of two image pairs; and *Post-Rotation View* predicts the resulting view after a text-specified rotation. Overall, the two-view tasks measure how precisely a model captures rotation, from detecting a change to specifying its exact magnitude; the multi-view tasks test how well it matches, compares, and applies rotation relations across observations.

Evaluation Protocol. We evaluate nine open-source and three proprietary VLMs zero-shot under fixed input/output formats, counting unparseable responses as incorrect. We report per-task accuracy, Macro-8 as the unweighted mean across tasks, and Micro-8 over all examples. Exact Angle (Open) uses normalized exact match, and human performance provides a reference point. Additional details on benchmark construction, task specifications, evaluation protocols, and random baselines are provided in Appendix A & B.

3.1 Overall Performance and Trends

Current VLMs show a sharp boundary in rotation understanding: they can reliably tell whether an object has rotated, but struggle to judge how far (Table 1). Across the nine open-source models, Rotation Detection averages 84.2%, with the strongest models approaching ceiling accuracy. Once rotation magnitude is required, however, performance falls close to chance: 90° Threshold and Coarse Angle average only 37.1% and 39.2%, respectively, against a 33.3% chance level, far below human accuracy (93.3% and 88.9%).

A similar pattern appears in the multi-image tasks. View Matching, which only requires identifying a named view such as front or left, averages 49.9%. Performance drops when models must compare rotation magnitudes (Rotation Comparison, 39.4%) and drops further when they must apply a rotation to predict a new view (Post-Rotation View, 28.5%). Proprietary models follow the same pattern at higher absolute performance. Across both two-view and multi-view tasks, models are stronger at detecting or matching rotations than at estimating or using rotation magnitude.

3.2 Canonical-Angle Collapse

Beyond accuracy, human errors still reflect a coherent estimate of object rotation. Although human performance on Exact Angle is also limited (56.5%), their errors are typically local, such as confusing 30° with 45°, while remaining consistent across granularities. VLM errors do not show this consistency. We closely examine three representative open-source models and find that they concentrate 91.8%, 98.4%, and 99.6% of their predictions on just three canonical angles ($0^\circ/90^\circ/180^\circ$). We call this behavior **canonical-angle collapse**, with 90° emerging as an especially dominant, often default, response (Figure 3). Closed-source models show a similar pattern (Appendix D).

Model	Exact Angle Task		
	MCQ	Open	MCQ (text-only)
Qwen3-VL-8B	91.8	97.1	97.5
InternVL3.5-8B	98.4	100.0	100.0
Qwen2.5-VL-7B	99.6	100.0	100.0

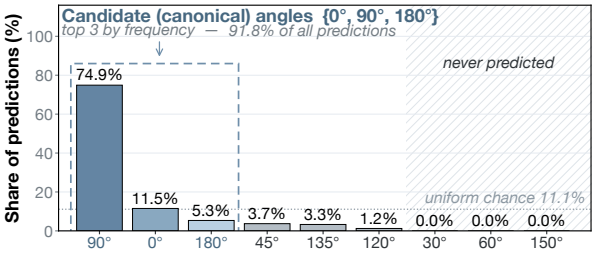


Figure 3: Canonical-angle collapse across input and response settings. **Left:** share of predictions in the canonical set $\{0°, 90°, 180°\}$; darker cells indicate stronger collapse. **Right:** Qwen3-VL-8B's Exact Angle (MCQ) predictions concentrate primarily at $90°$.

The collapse persists even without visual input: after removing all visual tokens, 97.5–100% of predictions still fall on the three canonical angles, indicating a strong textual prior toward these values. Visual input still matters, however: it changes 37% of predictions and raises accuracy from the 11.1% chance level to 22.2–24.7%. Yet this gain is concentrated almost entirely on $0°$, which reaches 100% recall and 89.0% precision. By contrast, $90°$ accounts for 67–75% of all predictions but attains only 12.9% precision, barely above its 11.1% base rate, while accuracy on noncanonical angles remains at 0–2.5%. This pattern cannot be explained simply by rounding to the nearest canonical angle, which would predict $0°$ for $30°$ or $180°$ for $150°$. Instead, the models can reliably separate zero from nonzero rotation, but once the rotation is nonzero, their exact-angle predictions carry little information about its magnitude.

4 Explaining the Rotation Reasoning Boundary

Why does rotation reasoning deteriorate once finer-grained rotation information is required? A natural explanation is that VLMs do not capture rotation magnitude from the visual input at all. To test this, we examine whether the relevant rotation information is present in the internal representations of three representative models. We focus on three increasingly fine-grained two-view tasks: Rotation Detection, Coarse Angle, and Exact Angle. We train linear RidgeClassifier probes (Hoerl & Kennard, 1970) on frozen vision encoder (VE) and LLM representations using five-fold object-disjoint cross-validation with out-of-fold (OOF) predictions (Belinkov, 2022). For comparison, we evaluate the model's own answer, i.e., its zero-shot prediction, on the same examples. Both are scored with reference-adjusted balanced accuracy (AdjBA; 0 = chance, 100 = perfect) (Brodersen et al., 2010). Full details are provided in Appendix E.

4.1 Probing the Vision Encoder

We first probe the vision encoder to test whether it captures rotation-magnitude information. Since the VE encodes the two images independently, we use the simplest absolute difference (`absdiff`), *i.e.*, $|f_1 - f_2|$ on their frozen representations f_1 and f_2. This operation is fixed and parameter-free: it adds no learnable relational module, so any rotation information recovered here is already carried by the two single-image representations.

We probe L8, L16, and the final VE layer. Across all three models, rotation information shows a clear coarse-to-fine decline in recoverability (Figure 4). Rotation Detection is nearly perfectly recoverable (98.0–98.6 AdjBA, averaged across layers), Coarse Angle remains substantially recoverable (28.0–30.2), and Exact Angle falls further (15.0–20.1). At many layers the probe is also more accurate than the model's own answer, most consistently for Coarse Angle, where the largest per-layer margin reaches $+24.7$, $+23.5$, and $+40.8$ AdjBA for the three models. As a sanity check, replacing the true Coarse Angle labels with random labels drops probe accuracy from 28.4–38.9 AdjBA to between -6.2 and -3.8 (Appendix E.6), indicating that the signal reflects genuine rotation information.

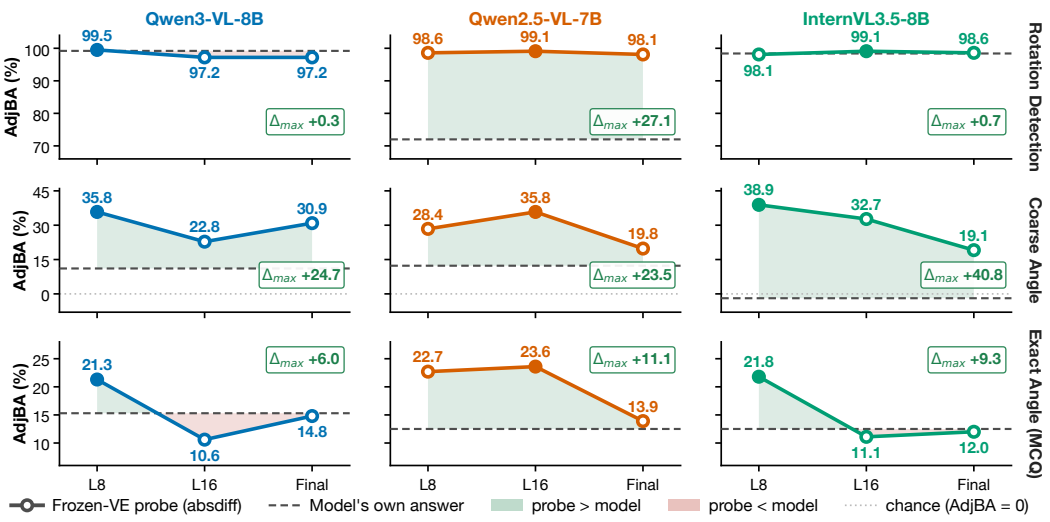


Figure 4: Linear probe accuracy at L8, L16, and the final VE layer, with the model's own answer as a reference. Recoverability declines from Rotation Detection to Coarse Angle to Exact Angle.

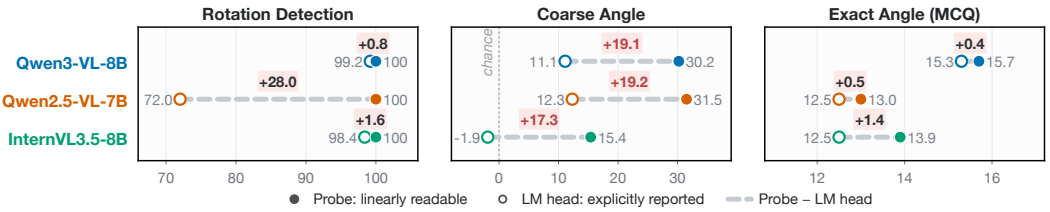


Figure 5: Probe vs. LM head at h_{pre}. Probe (filled) and LM head (open) are two linear heads reading the same pre-answer representation. The gap is largest on Coarse Angle.

4.2 Probing at the Output Head

We next probe h_{pre}, the final-layer hidden state at the last prompt position, from which the LM head produces the model's own answer. Unlike the VE analysis, we apply no `absdiff` or other external relational operation. The probe and the LM head are therefore two linear heads reading the same representation, which makes the comparison direct at the representation level.

At h_{pre} we observe a similar coarse-to-fine decline as in the VE: across the three models, probes reach 100.0 AdjBA on Rotation Detection, 15.4–31.5 on Coarse Angle, and 13.0–15.7 on Exact Angle. The rotation relation between the two views is therefore already encoded in the model's own representation, and remains linearly readable there. The probe again outperforms the LM head, most clearly on Coarse Angle, where it leads by 17.3–19.2 AdjBA (Figure 5). Rotation Detection is already near ceiling, and on Exact Angle the two remain close, differing by only 0.4–1.4 AdjBA.

The models' failure at finer granularities thus reflects two complementary limitations: as the angle becomes finer, the recoverable information weakens, and what remains is not fully exploited at readout. On Rotation Detection the information is almost perfectly recoverable and the model generally reports it correctly. On Coarse Angle recoverability drops but stays substantial, yet the gap to the LM head is largest, so the available information is under-used. On Exact Angle both the information and the answer are weak: the model reliably resolves only whether the rotation is zero, and elsewhere falls back on its textual prior, consistent with the canonical-angle collapse observed earlier (Section 3.2).

This gap is not specific to Coarse Angle task or to how the probes are trained: it also appears on 90° Threshold, another coarse-granularity task using different images and prompts, and persists when the probes are trained on an external dataset (Appendix E.5).

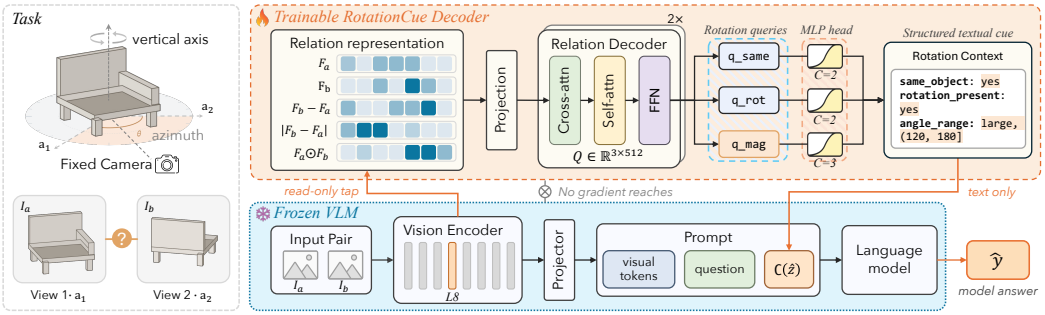


Figure 6: **RotationCue.** The relation representation is read from vision layer L8 and passed to a lightweight decoder, without modifying the frozen VLM. Three learnable rotation queries predict a three-field rotation context, which is reinjected into the VLM as text only.

5 RotationCue: Rotation Context from Frozen Representations

The probing results show that VLM representations retain substantial rotation information, particularly at coarse angular granularities. This raises a natural question: can this information be further utilized to improve object rotation reasoning? We introduce **RotationCue**, which uses a lightweight decoder to recover rotation information from internal representations as structured context, and reinjects it into the frozen VLM for reasoning. All rotation information supplied to the model is decoded from its own frozen representations. RotationCue thus extends the probing analysis from diagnosis to improvement.

5.1 Recovery and Reinjection

RotationCue reads rotation information directly from the frozen visual representations, before language generation. Guided by the probing results in Section 4.1, we tap L8, where coarse rotation information remains reliably recoverable across all three models. Given two views I_A and I_B, we extract $F_A = V^{(8)}(I_A)$ and $F_B = V^{(8)}(I_B)$. To recover richer rotation information from the two views, we construct the following pairwise relation representation:

$$R_{\text{full}}(F_A, F_B) = [F_A; F_B; F_B - F_A; |F_A - F_B|; F_A \odot F_B].$$

Structured rotation context. RotationCue recovers three rotation-related fields as structured context: `angle_range`, `same_object`, and `rotation_present`. The primary field, `angle_range`, represents rotation magnitude as `small` $(0°, 60°)$, `medium` $[60°, 120°]$, or `large` $(120°, 180°]$. We choose this granularity because Coarse Angle retains substantial recoverable information while showing the largest gap from the model's own predictions. The other two fields are conditions: `same_object` determines whether the views depict the same object, and `rotation_present` determines whether a rotation occurs. The angle range is provided only when both hold, preventing irrelevant rotation context from being injected.

R_{full} is projected into a low-dimensional space and aggregated by three learnable rotation queries. Three heads predict `same_object` → `rotation_present` → `angle_range`, with an angular-distance loss penalizing larger cross-bin errors. The decoded outputs $\hat{z} = D_\phi(R_{\text{full}})$ are serialized with a fixed template and reinjected as context: $\hat{y} = M(I_A, I_B, q, C(\hat{z}))$. Only the decoder is trained; the base VLM remains frozen. Full architectural, serialization, and training details are provided in the Appendix F. Training uses only external multi-view data: ALOI-View (Geusebroek et al., 2005) and SelectiveMV ModelNet40v1 (Alzahrani et al., 2024; Wu et al., 2015), with object-disjoint train/val splits. OR-Bench is held out entirely for evaluation. The same readout layer, relation representation, decoder design, and training protocol are used across all three models, with one decoder per model serving all tasks.

5.2 Results on OR-Bench

RotationCue improves all 24 model–task combinations across three models and eight tasks, with Macro-8 gains of 11.2, 12.6, and 7.9 percentage points for Qwen3-VL-8B, Qwen2.5-VL-7B, and

Table 2: Task-level RotationCue results on OR-Bench across three VLMs. Main values are scores (%); smaller lower-right values in parentheses are gains over Raw (Δ, percentage points), with shading encoding Δ. Mean Δ averages the gain across the three models.

Level	Task	Qwen3-VL-8B	Qwen2.5-VL-7B	InternVL3.5-8B	Mean Δ
Two-view	Rotation Detection	100.0 (+0.4)	99.0 (+13.0)	100.0 (+0.8)	+4.7
	Coarse Angle	66.7 (+26.0)	63.4 (+21.8)	60.5 (+28.4)	+25.4
	90° Threshold	55.6 (+18.6)	57.2 (+19.8)	44.0 (+6.6)	+15.0
	Exact Angle (MCQ)	32.9 (+8.2)	33.7 (+11.5)	28.0 (+5.8)	+8.5
	Exact Angle (Open)	34.6 (+9.1)	27.2 (+4.1)	27.2 (+3.7)	+5.6
Multi-view	View Matching	67.6 (+2.3)	57.4 (+1.4)	50.9 (+1.8)	+1.8
	Rotation Comparison	57.2 (+8.6)	58.4 (+16.8)	57.2 (+13.2)	+12.9
	Post-Rotation View	49.0 (+16.2)	42.7 (+12.5)	39.1 (+2.6)	+10.4
Macro-8 (all tasks)		58.0 (+11.2)	54.9 (+12.6)	50.9 (+7.9)	+10.5

Table 3: Macro-8 (%) with (a) different intermediate-context strategies and (b) RotationCue content quality. Self-gen uses the same three-field format as RotationCue; Shuffled uses a cue from another example; Oracle uses ground-truth rotation information.

	(a) Source of intermediate context					(b) RotationCue content quality			
Model	Raw	ZS-CoT	5-shot	Self-gen.	RotationCue	Shuffled	Raw	RotationCue	Oracle
Qwen3-VL-8B	46.8	47.1	52.3	48.6	**58.0**	44.4	46.8	58.0	**65.3**
Qwen2.5-VL-7B	42.3	43.1	44.4	47.1	**54.9**	38.7	42.3	54.9	**63.6**
InternVL3.5-8B	43.0	39.7	44.4	47.4	**50.9**	42.9	43.0	50.9	**59.6**

InternVL3.5-8B, respectively. With RotationCue, Qwen3-VL-8B reaches 58% Macro-8, surpassing GPT-5.4 at 54.4%. Full task-level results are reported in Table 2. The largest gains occur on tasks directly aligned with the recovered rotation cue. Coarse Angle improves by 21.8–28.4 points. Rotation Detection reaches 100.0% for Qwen3-VL-8B and InternVL3.5-8B, while Qwen2.5-VL-7B improves from 86.0% to 99.0%. Importantly, the gains extend beyond tasks directly specified by the cue. Although `angle_range` provides only three coarse angular intervals, Exact Angle (MCQ) improves by 5.8–11.5 points, Rotation Comparison by 8.6–16.8 points, and Post-Rotation View by 2.6–16.2 points, with consistent gains also on Exact Angle (Open) and the 90° Threshold task.

We attribute these gains to RotationCue acting as an explicit textual anchor: it turns rotation information from the model's own representations into text, bridging what the model encodes visually and what it can use in language reasoning. RotationCue also substantially mitigates canonical-angle collapse, reducing the canonical-angle prediction rate from 91.8%–99.6% to 59.3%–78.6%. Taking a closer look at Qwen3-VL-8B on Exact Angle (MCQ), we find that when the cue is correct, accuracy on non-zero rotations rises from 19.4% to 40.3% on the same items, and the prediction distribution becomes more plausible.

5.3 Context Source and Quality

We further compare RotationCue with alternative strategies for supplying intermediate context (Table 3). Besides CoT-based methods (zero-shot CoT and 5-shot CoT), we prompt the model to generate the same three-field context used by RotationCue before producing its final answer (Self-gen), which holds the format fixed and changes only where the content comes from. RotationCue performs best on all three models, exceeding 5-shot CoT by 5.7, 10.4, and 6.5 Macro-8 points. Free-form reasoning alone brings little benefit: Zero-shot CoT brings almost no improvement over Raw and even hurts performance on InternVL3.5-8B, while the more structured Self-gen yields consistently better results.

We then keep the three-field structure fixed and vary only the quality of its content. Performance consistently follows Shuffled $<$ Raw $<$ RotationCue $<$ Oracle. Shuffled cues lower Macro-8 by 0.1–3.5 points, while Oracle yields a further 7.4–8.7-point gain over RotationCue. Together, these

Table 4: Comparison between RotationCue and Rotation-SFT on Qwen3-VL-8B (%). RotationCue achieves larger gains on OR-Bench while largely preserving performance on external benchmarks. Parenthesized values denote changes from the raw model, computed from displayed scores.

	OR-Bench		External benchmarks				
Method	**Macro-8**	**Micro-8**	**VSR-ZS**	**CV 2D**	**CV 3D**	**CV Avg.**	**SpinBench**
Qwen3-VL-8B (Raw)	46.8	53.0	83.6	80.1	93.3	86.7	67.0
Rotation-SFT	51.2 (+4.4)	56.3 (+3.3)	75.6 (−8.0)	75.4 (−4.7)	89.0 (−4.3)	82.2 (−4.5)	61.8 (−5.2)
RotationCue	**58.0 (+11.2)**	**62.9 (+9.9)**	84.0 (+0.4)	80.1 (—)	92.5 (−0.8)	86.3 (−0.4)	**69.2 (+2.2)**

results show that RotationCue's advantage comes from both structured, task-specific intermediate context and more accurate rotation information.

RotationCue is also more efficient and stable than free-form CoT. Under the same inference setup, it adds only 1.2% latency over Raw while being about $2.4\times$ faster than zero-shot CoT. Its intermediate context is compact, and under stochastic sampling it produces more consistent predictions than free-form CoT while maintaining a clear accuracy advantage.

5.4 Why Not Just Fine-Tune the Base Model?

RotationCue keeps the base VLM frozen and places rotation supervision externally, in a lightweight decoder that recovers rotation information from its representations. An alternative is to apply comparable supervision internally, by updating trainable parameters within the VLM itself. Rotation-SFT does this: it uses the same external multi-view sources and angular semantics, reformulated as visual question–answer pairs, and fine-tunes the language side through LoRA while keeping the vision encoder and projector frozen. Training details and per-task results are provided in the Appendix H.

Rotation-SFT underperforms RotationCue on both aggregate metrics, scoring 51.2 versus 58.0 on Macro-8 and 56.3 versus 62.9 on Micro-8, corresponding to gaps of 6.8 and 6.6 points. Its gains are also concentrated: Rotation-SFT performs strongest on Coarse Angle, the task most directly aligned with its supervision, while showing no consistent improvement across the remaining OR-Bench tasks and even degrading on several of them.

This specialization also affects general performance. Relative to the base model, Rotation-SFT drops by 8.0 points on VSR-ZS (Liu et al., 2023), 4.5 on CV-Bench (Tong et al., 2024), and 5.2 on SpinBench (Zhang et al., 2026). RotationCue leaves the base model unchanged and largely preserves its performance on these benchmarks, and requires only 0.23 A100 GPU-hours to train, compared with roughly 42 A100 GPU-hours for Rotation-SFT.

6 Conclusion

Object rotation is a basic form of visual change, and current VLMs handle it in a strikingly uneven way: they reliably register that an object has turned, but not how far. OR-Bench makes this failure precise, showing that it is not a gradual loss of accuracy with finer granularity but a break at the point where a magnitude must be produced at all. Representation probing then locates it in two places: recoverable rotation information weakens as granularity becomes finer, and the readout leaves unused much of what remains. RotationCue puts that unused information to work, recovering rotation state from frozen representations and returning it to the model as context. Across three different VLMs, it improves object-rotation reasoning on every OR-Bench task, without updating a VLM weight or compromising general capability.

AI use statement

Generative AI tools were used to assist with manuscript drafting, translation, and language polishing.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Mona Alzahrani, Muhammad Usman, Saeed Anwar, and Tarek Helmy. Selective multi-view deep model for 3d object classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 728–736, 2024. doi: 10.1109/CVPRW63382.2024.00077.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *Proceedings of the 20th International Conference on Pattern Recognition*, pp. 3121–3124. IEEE, 2010. doi: 10.1109/ICPR.2010.764.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F. Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. ABO: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21126–21136, 2022.

Duolikun Danier, Mehmet Aygün, Changjian Li, Hakan Bilen, and Oisin Mac Aodha. Depthcues: Evaluating monocular depth perception in large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20049–20059, June 2025.

Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: VLMs overlook their visual representations. In *Conference on Language Modeling (COLM)*, 2025.

Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005. doi: 10.1023/B:VISI.0000042993.50813.60.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9161–9175, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.568.

Raphi Kang, Hongqiao Chen, Georgia Gkioxari, and Pietro Perona. Linear mechanisms for spatiotemporal reasoning in vision language models, 2026. URL https://arxiv.org/abs/2601.12626.

Fei Kong, Jinhao Duan, Kaidi Xu, Zhenhua Guo, Xiaofeng Zhu, and Xiaoshuang Shi. LRR-Bench: Left, right or rotate? vision-language models still struggle with spatial understanding tasks. *arXiv preprint arXiv:2507.20174*, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. doi: 10.1162/tacl_a_00566.

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

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, May 2024a. Accessed: 2026-08-04.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, July 2024b. Accessed: 2026-08-04.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, March 2026. Accessed: 2026-08-04.

Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

Alexa R. Tartaglini, Satchel Grant, Daniel Wurgaft, Christopher Potts, and Judith E. Fan. Diagnosing bottlenecks in data visualization understanding by vision-language models. *arXiv preprint arXiv:2510.21740*, 2025.

Nikos Theodoridis, Reenu Mohandas, Ganesh Sistu, Anthony Scanlan, Ciarán Eising, and Tim Brophy. Probing visual concepts in lightweight vision-language models for automated driving. *arXiv preprint arXiv:2603.06054*, 2026.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024. doi: 10.52202/079017-2771.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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

Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M. de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6d spatial reasoning of large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24669–24679, 2025b. doi: 10.1109/CVPR52734.2025.02297.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015. doi: 10.1109/CVPR.2015.7298801.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. doi: 10.1109/CVPR52734.2025.00994.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. SpinBench: Perspective and rotation as a lens on spatial reasoning in VLMs. In *International Conference on Learning Representations*, 2026.

Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A OR-Bench Construction and Composition

OR-Bench evaluates object-level rotation reasoning through eight tasks: five two-view tasks assess the same rotation relation at different output granularities, and three multi-view tasks require models to match, compare, or apply rotation relations across observations. The benchmark contains 2,109 test examples constructed from 73 objects.

A.1 Source Data and Rotation Labels

OR-Bench uses the 360° turntable subset of Amazon Berkeley Objects (ABO) (Collins et al., 2022). Each product scan contains 72 views captured at 5° azimuthal intervals. The target rotation is the unsigned shortest azimuthal displacement between two views of the same object rotated about its vertical axis, rather than an in-plane image rotation.

For frame indices $i, j \in \{0, \ldots, 71\}$, let $k = |i - j|$. The relative rotation angle is

$$\theta = 5^\circ \min(k, 72 - k) \in [0^\circ, 180^\circ].$$

This is equivalent to the azimuth-based definition in Section 3. Frame offsets therefore provide exact rotation labels without pose estimation or manual angle annotation.

The target describes the angular separation between endpoint views. When a task requires rotation direction, as in Post-Rotation View, the direction is provided explicitly in the text.

A.2 Object Screening

We remove objects whose orientation is visually ambiguous due to symmetry around the vertical axis. For these objects, different azimuths can produce nearly indistinguishable views despite having different rotation labels. Screening retains objects with distinguishable orientation cues, yielding 73 objects for benchmark construction. Figure 7 illustrates the screening criterion.



Figure 7: **Illustrative examples of object screening.** Objects whose orientation is visually ambiguous due to symmetry, such as the vase (left), are excluded. Objects with distinctive orientation cues, such as the sofa (right), are retained.

Table 5: **Composition of OR-Bench.** Uniform reference denotes uniform guessing over the answer choices for closed-set tasks and over the nine reference angles for Exact Angle (Open).

Task Group	Task	Examples	Imgs./Ex.	Uniform reference
Two-view	Rotation Detection	486	2	50.0%
	Coarse Angle	243	2	33.3%
	90° Threshold	243	2	33.3%
	Exact Angle (MCQ)	243	2	11.1%
	Exact Angle (Open)	243	2	11.1%
Multi-view	View Matching	216	4	33.3%
	Rotation Comparison	243	4	33.3%
	Post-Rotation View	192	5	25.0%

A.3 Reference Angles and Task Organization

Tasks are constructed around nine reference angles:

$$\{0°, 30°, 45°, 60°, 90°, 120°, 135°, 150°, 180°\}.$$

These angles provide a symmetric set of representative rotations spanning the full $0°$–$180°$ range, including canonical angles and common intermediate angles.

Two-view tasks. Five tasks query the rotation relation within a single image pair. *Rotation Detection* asks whether the orientation changes. *Coarse Angle* assigns the rotation to one of three intervals: small ($[0°, 60°)$), medium ($[60°, 120°]$), or large ($(120°, 180°]$). $90°$ *Threshold* distinguishes rotations below, equal to, or above $90°$. *Exact Angle (MCQ)* requires selecting the exact angle, while *Exact Angle (Open)* requires predicting it directly.

Coarse Angle and the two Exact Angle variants use the same image pairs. Each pair is associated with a coarse interval label and an exact angle label, with the latter queried in both multiple-choice and open-response formats. This shared-input design enables comparison across output granularities and response formats while holding the visual input fixed.

Multi-view tasks. Three tasks require rotation relations to be used across multiple observations. *View Matching* identifies a specified object view, such as front, back, left, or right, from candidate images. *Rotation Comparison* compares the rotation magnitudes of two image pairs. *Post-Rotation View* selects the resulting view after a text-specified rotation. Together, these tasks assess whether models can use rotation relations for view matching, magnitude comparison, and transformation prediction.

A.4 Composition and Balancing

The eight tasks contain 2,109 test examples from 73 objects. Because source images can be reused across examples, these examples contain 5,712 image occurrences drawn from 2,392 distinct source images. Closed-set tasks balance target classes and correct answer positions. For image-selection tasks, all candidate images depict the same object, preventing object identity from serving as a shortcut. Table 5 summarizes the benchmark composition.

B Evaluation Protocol and Task Definitions

We specify the evaluation protocol and illustrate the eight OR-Bench tasks, grouped into five two-view tasks and three multi-view tasks. Rotation angles denote unsigned shortest azimuthal changes about the object's vertical axis; direction is provided explicitly in the text when required. Each display follows the image order, question format, and answer space used during evaluation. For closed-set tasks, target classes and correct answer positions are balanced, with option order varied where applicable to limit positional bias. All reported results are computed on the complete task sets summarized in Table 5.

B.1 Scoring and Decoding

Evaluation protocol. We evaluate nine open-source and three proprietary VLMs zero-shot under fixed input/output formats. For closed-set tasks, responses are parsed against the option letters allowed by each instance and scored by exact agreement with the stored answer. Unparseable responses are recorded as `UNKNOWN` and counted as incorrect. For Exact Angle (Open), numeric values are extracted and degree formatting is normalized; the score requires exact agreement with the ground-truth angle. We report per-task accuracy, Macro-8 as the unweighted mean of the eight task accuracies, and Micro-8 as accuracy over all examples.

Decoding. All benchmark evaluations use greedy decoding (`do_sample=false`) with at most 32 new tokens. Temperature and top-p are unset when sampling is disabled. Each model is evaluated once on the full benchmark; the reported results are single-run measurements.

B.2 Two-view Tasks

Five tasks query the rotation relation within a single image pair at different output granularities.

Rotation Detection



Image 1



Image 2

The camera is fixed and the object rotates in place.

Do the two images show the object in the same orientation, or has it rotated?

A. Rotated
B. Same orientation

Only answer with a single capital letter from (A, B).

Setup. The label is *Rotated* iff $\theta > 0^\circ$; same-orientation examples reuse the identical frame. Target classes and correct answer positions are balanced.

Coarse Angle



Image 1



Image 2

The camera is fixed and the object rotates in place.

Which coarse interval best describes the rotation from Image 1 to Image 2?

A. $[0°, 60°)$
B. $[60°, 120°]$
C. $(120°, 180°]$

Only answer with a single capital letter from (A, B, C).

Setup. The three intervals contain $\{0°, 30°, 45°\}$, $\{60°, 90°, 120°\}$, and $\{135°, 150°, 180°\}$, respectively. Target classes and correct answer positions are balanced; option order varies across examples.

$90°$ Threshold



Image 1



Image 2

The camera is fixed and the object rotates in place.

Is the unsigned rotation angle from Image 1 to Image 2 less than, equal to, or greater than $90°$?

A. Greater than $90°$
B. Equal to $90°$
C. Less than $90°$

Only answer with a single capital letter from (A, B, C).

Setup. The three target classes are $\theta < 90°$, $\theta = 90°$, and $\theta > 90°$. Target classes and correct answer positions are balanced; option order varies across examples.

Exact Angle (MCQ)



Image 1



Image 2

The camera is fixed and the object rotates in place.

How many degrees did the object rotate from Image 1 to Image 2?

A. 30° B. 45° C. 60° D. 90° E. 120° F. 135° G. 150° H. 180° I. 0°

Only answer with a single capital letter from (A, B, C, D, E, F, G, H, I).

Setup. All nine reference angles occur equally often. Option order varies across examples, and correct answer positions are balanced.

Exact Angle (Open)



Image 1 Image 2

The camera is fixed and the object rotates in place.

How many degrees did the object rotate from Image 1 to Image 2?

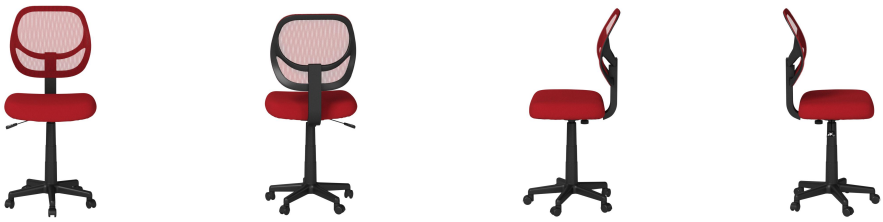
Answer with a single number in degrees.

Setup. Exact Angle (Open) removes the candidate list and requires a direct numeric prediction. The score requires exact agreement with the ground-truth angle after normalization.

B.3 Multi-view Tasks

Three tasks require models to match, compare, and apply rotation relations across multiple observations.

View Matching



Reference Candidate **A** Candidate **B** Candidate **C**

The reference image shows the object from the **Front** view. Select the candidate image that best shows the object from the **Left** view.

Only answer with a single capital letter from (A, B, C).

Setup. Reference and target cardinal views vary across examples. All candidates depict the same object as the reference, and correct answer positions are balanced.

Rotation Comparison

The camera is fixed. Each pair shows the same object before and after an in-place rotation.

Pair A **Pair B**



Image 1 Image 2 Image 1 Image 2

Which pair has the larger unsigned rotation angle?

A. The two pairs have the same rotation angle.
B. Pair A has the larger rotation.
C. Pair B has the larger rotation.

Only answer with a single capital letter from (A, B, C).

Setup. The target is determined directly by comparing θ_A and θ_B. All four images depict the same physical object, preventing category or object identity from serving as a shortcut. The three comparison outcomes and correct answer positions are balanced.

Post-Rotation View

Study the reference image, then mentally simulate the object rotating in place.



Reference

Imagine the object rotates 150° **clockwise** while the camera stays fixed. Clockwise and counterclockwise are defined from a top-down view.

After imagining this rotation, which candidate image matches the final view?



Candidate A Candidate B Candidate C Candidate D

Only answer with a single capital letter from (A, B, C, D).

Setup. Rotation direction is explicitly specified in the instruction. All candidates depict the same object and are sampled from the instructed-direction trajectory. Duplicate candidate images and opposite-direction, same-magnitude distractors are excluded, and correct answer positions are balanced.

B.4 Output-Format Controls

We perform two checks for output-format artifacts. First, all 486 Rotation Detection responses from each of Qwen2-VL-7B, LLaVA-OneVision-Qwen2-7B, and Cosmos-Reason1-7B are successfully parsed, despite accuracies of 62.8%, 60.5%, and 65.0%, respectively. Second, the option preferred by the LM head agrees with the option recovered from the final generated response in 100% of

evaluated cases. In these checks, errors are not explained by response parsing or a mismatch between native option readout and the emitted answer.

C Human Reference Evaluation

Human performance is measured on a stratified subset of OR-Bench as a reference for interpreting model performance. Five participants each complete 628 sampled examples, yielding 3,140 participant–item responses in total. The subset covers all eight tasks: 96 examples for Rotation Detection, 81 for Coarse Angle, 72 for the 90° Threshold, 81 each for Exact Angle (MCQ) and Exact Angle (Open), 72 for View Matching, 81 for Rotation Comparison, and 64 for Post-Rotation View. Sampling is stratified by task-relevant variables, including rotation angle, target class, comparison outcome, and correct answer position, to provide balanced coverage of each task.

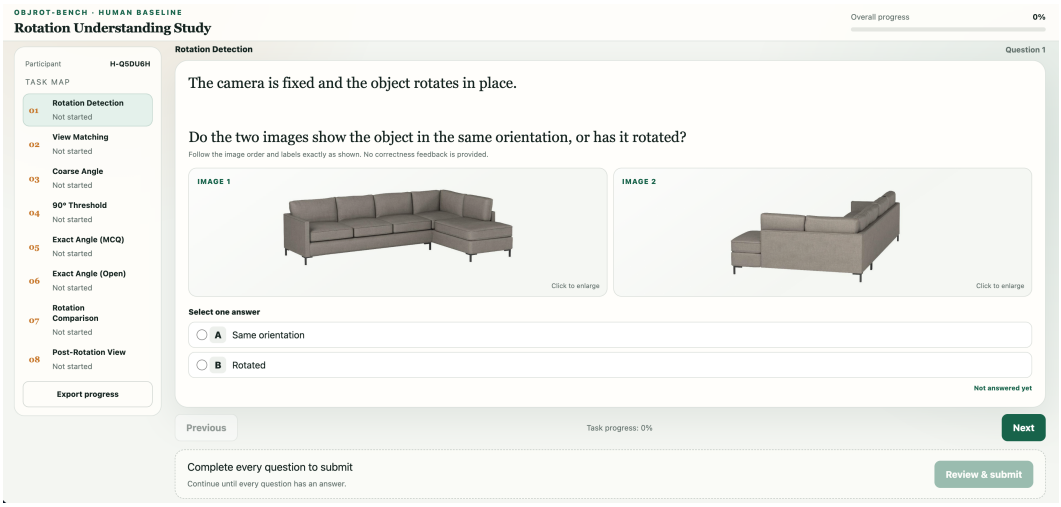


Figure 8: **Human evaluation interface for OR-Bench.** The interface supports task-wise navigation and progress tracking. A Rotation Detection example is shown.

Evaluation is divided into four sessions without correctness feedback. Participants receive the same images, task instructions, questions, and answer spaces used for model evaluation. Human responses are scored using the same task-level criteria as model outputs, including normalized exact match for Exact Angle (Open). Human Macro-8 is the unweighted mean of accuracy across the eight tasks.

D Canonical-Angle Collapse in Proprietary Models

Proprietary models also concentrate their predictions on the canonical angles $\{0°, 90°, 180°\}$, consistent with the pattern described in the main text (Table 6). The most frequently predicted angle is 90° for all three models.

Table 6: Share of predictions assigned to canonical angles and to 90° by proprietary models (%).

Model	$\{0°, 90°, 180°\}$	$90°$
GPT-4o	82.0	62.3
GPT-4o-mini	98.4	70.5
GPT-5.4	68.9	28.7

E Representation Diagnostics: Probe Design and Implementation

We use linear probes to measure task-relevant rotation information recoverable from frozen VLM representations. The main analysis covers three increasingly fine-grained tasks: Rotation Detection,

Coarse Angle, and Exact Angle. We specify the task alignment, object-held-out protocol, evaluation metric, and two readout locations, followed by the additional 90° Threshold results and random-label control.

E.1 Task Alignment and Object-Held-Out Protocol

We probe the representations using the examples and labels of the corresponding OR-Bench tasks: Rotation Detection (486 examples), Coarse Angle (243), and Exact Angle (243). These tasks progressively increase the angular resolution required by the readout.

All probes use five-fold object-disjoint cross-validation. Let $\mathcal{O}^{(k)}_{\text{train}}$ and $\mathcal{O}^{(k)}_{\text{test}}$ denote the training and test objects in fold k. We enforce

$$\mathcal{O}^{(k)}_{\text{train}} \cap \mathcal{O}^{(k)}_{\text{test}} = \varnothing.$$

Each example receives exactly one out-of-fold prediction, and all preprocessing and classifier fitting use only the corresponding training fold.

E.2 Evaluation Metric

We compute balanced accuracy (BA):

$$\text{BA} = \frac{1}{C}\sum_{c=1}^{C} \text{Recall}_c,$$

where C is the number of classes. BA gives equal weight to each class.

To account for the different uniform-reference levels across tasks, we report reference-adjusted balanced accuracy:

$$\text{AdjBA} = 100 \times \frac{\text{BA} - q}{1 - q}, \qquad q = \frac{1}{C}.$$

Under this convention, $\text{AdjBA} = 0$ corresponds to uniform guessing and $\text{AdjBA} = 100$ to perfect classification; negative values indicate performance below the uniform reference. This places the 2-way, 3-way, and 9-way probes on a common reference-adjusted scale.

E.3 Vision-Encoder Readout

We examine three depths of the frozen vision encoder: L8, L16, and the final layer. For each image pair, tokens associated with Image 1 and Image 2 are separated and independently mean-pooled:

$$f_1 = \text{MeanPool}(F_1), \qquad f_2 = \text{MeanPool}(F_2).$$

Mean pooling introduces no learned aggregation parameters.

For the primary vision-encoder (VE) probe, the pair is represented by the parameter-free absolute difference:

$$|f_1 - f_2|.$$

This representation is symmetric to image order and aligned with the unsigned rotation quantities used by the probe tasks. It introduces no learnable relational module, so the recovered rotation information is carried by the frozen single-image representations.

The readout uses feature standardization followed by a linear RidgeClassifier:

$$|f_1 - f_2| \longrightarrow \text{StandardScaler} \longrightarrow \text{RidgeClassifier}(\alpha = 10).$$

The scaler and classifier are fitted independently within each training fold, while all VLM parameters remain frozen.

We additionally evaluate signed difference, concatenation, and richer pair representations as diagnostic variants. These serve as robustness checks; the primary VE analysis uses absolute difference throughout.

E.4 Matched Readout at the Native Answer Interface

To test whether the VLM itself has constructed a two-view relation, we probe the state immediately consumed by its native LM head.

We denote by h_{pre} the final post-normalization hidden state used to predict the first answer token. At this location, no external absolute difference, pooling operation, or pair representation is introduced. The probe therefore tests whether the two-view rotation relation is linearly readable from the model's own representation.

Both readouts operate on the same hidden state. Native prediction uses the frozen vocabulary head:

$$\ell_{\mathrm{native}} = W_{\mathrm{vocab}} h_{\mathrm{pre}},$$

whereas the task-aligned probe uses fitted linear weights:

$$\ell_{\mathrm{probe}} = W_{\mathrm{probe}} h_{\mathrm{pre}}.$$

The comparison tests whether task-relevant rotation information reaches the native answer interface and whether the pretrained readout effectively uses it.

E.5 Additional Probing Results: Task and Protocol Generalization

We report two robustness checks on the coarse-granularity readout gap: one varying the task, and one varying the probe training protocol.

A different task and question form. The 90° Threshold task uses a separate image set and a different question form from Coarse Angle. All three models answer at chance on it (5.6–6.2 AdjBA). Linear probes nonetheless recover the rotation relation well above chance at every VE layer, and at h_{pre} the probe exceeds the model's own answer for all three models (Table 7). The readout gap therefore reappears under a different image set and question form, here on a task that no model solves.

Table 7: Probe and native performance on 90° Threshold (AdjBA, %; chance-adjusted, chance $= 0$). VE probes use `absdiff` on the two frozen single-image representations; the h_{pre} probe and the LM head read the same representation. *Native* is the model's own answer and Δ is the probe–native difference at h_{pre}. Evaluated on 243 image pairs.

	VE probes			Output-head readout		
Model	L8	L16	Final	h_{pre}	LM Head	Δ
Qwen3-VL-8B	38.4	22.2	13.4	27.2	5.6	+21.6
Qwen2.5-VL-7B	29.2	31.0	13.9	9.9	6.2	+3.7
InternVL3.5-8B	31.0	20.8	24.1	24.1	6.2	+17.9

A different probe training protocol. To verify that the probes do not rely on idiosyncrasies of the evaluation set, we refit them entirely on external data. For each frozen model and each task we train an independent three-way linear probe on 972 ALOI image pairs, with 243 pairs held out for validation, object-disjoint splits, and angle, class and option distributions matched to the target task. We use the native task prompt, extract h_{pre} at the last prompt position, standardize with training-set statistics, and fit a `RidgeClassifier` whose penalty is selected on the external validation set only. The OR-Bench test sets (243 pairs each) are used for evaluation and never for fitting or model selection.

Under this protocol the probe again exceeds the model's own answer on both tasks and all three models (Table 8). Since the probe is trained on a different image distribution, these values are a lower bound on the linearly available rotation information.

Across both checks, the gap holds in every model and task examined: more rotation information is linearly present in the representation than the model's own answer reports.

Table 8: Probes trained on external ALOI data and evaluated on OR-Bench (AdjBA, %; chance-adjusted, chance = 0). Probes are fit on 972 ALOI pairs with object-disjoint splits and never see the test set. Δ is the probe–native difference. Each test set contains 243 image pairs.

	Coarse Angle			90° Threshold		
Model	Probe	Native	Δ	Probe	Native	Δ
Qwen3-VL-8B	22.2	11.1	+11.1	32.1	5.6	+26.5
Qwen2.5-VL-7B	16.0	12.3	+3.7	21.0	6.2	+14.8
InternVL3.5-8B	21.0	-1.9	+22.8	14.2	6.2	+8.0

E.6 Random-Label Control

We perform a random-label control on Coarse Angle using the VE L8 representations. For each fold, we keep the object split, frozen features, preprocessing, and Ridge configuration unchanged, but randomly permute the labels within the training partition:

$$\tilde{y}_{\text{train}} = \pi(y_{\text{train}}),$$

where π is a random permutation. Held-out labels remain unchanged. Randomizing the training labels reduces AdjBA below the uniform reference for all three models, supporting that the recovered signal reflects rotation information associated with the true labels.

Table 9: Random-label control for Coarse Angle readout from VE L8 representations (AdjBA, %).

Model	**True labels**	**Random labels**
Qwen3-VL-8B	35.8	−6.2
Qwen2.5-VL-7B	28.4	−4.3
InternVL3.5-8B	38.9	−3.8

F RotationCue: Recovery, Training, and Intervention Details

RotationCue recovers a structured rotation state from frozen internal visual representations and reinjects that state as context for the original VLM. No parameter of the base VLM is updated at any stage. For two images I_A and I_B, we independently extract mean-pooled Vision L8 representations,

$$F_A = \text{MeanPool}\left(V^{(8)}(I_A)\right), \qquad F_B = \text{MeanPool}\left(V^{(8)}(I_B)\right).$$

An external decoder D_ϕ predicts the rotation state,

$$\hat{z} = D_\phi\big(R_{\text{full}}(F_A, F_B)\big),$$

which is serialized as fixed structured context $C(\hat{z})$ and supplied to the unchanged base model,

$$\hat{y} = M_\theta\big(I_A, I_B, q, C(\hat{z})\big), \qquad \theta \text{ fixed.}$$

Thus, RotationCue uses the VLM only as a source of frozen internal representations and as the final reasoner; only the decoder parameters ϕ are trained.

The section proceeds from the predicted state and frozen visual inputs to the decoder architecture and hierarchical objective. It then documents the independent training data, shared optimization protocol, multi-image pairing, state-injection controls, and robustness evaluation.

F.1 Structured Rotation State

RotationCue predicts a hierarchical state

$$z = (z_s, z_r, z_c),$$

containing three fields. The first,

$$z_s \in \{\texttt{yes}, \texttt{no}\},$$

indicates whether the two images depict the same object. Conditional on $z_s = \texttt{yes}$,

$$z_r \in \{\texttt{yes}, \texttt{no}\}$$

indicates whether its orientation changes. When rotation is present,

$$z_c \in \{\texttt{small}, \texttt{medium}, \texttt{large}\}$$

describes its coarse magnitude,

$$\texttt{small} : (0^\circ, 60^\circ), \qquad \texttt{medium} : [60^\circ, 120^\circ], \qquad \texttt{large} : (120^\circ, 180^\circ].$$

The state follows the dependency

$$z_s \to z_r \to z_c.$$

If the images depict different objects, the rotation relation is undefined. When no rotation occurs, `angle_range` is set to `not_applicable`. For same-object pairs, `rotation_present=no` corresponds to $\theta = 0^\circ$. Together, this no-rotation state and the positive `small` angle range cover the benchmark's `small` class $[0^\circ, 60^\circ)$.

We recover a coarse intermediate state because Coarse Angle retains substantial recoverable information while showing the largest gap from the model's own predictions. RotationCue supplies this state as context for the original VLM to combine with the images and task requirements.

F.2 Frozen Visual Representation and Pair Construction

The two images are encoded independently. RotationCue uses only frozen Vision L8 representations and does not use language-model hidden states.

From F_A and F_B, we construct the fixed pairwise relation representation

$$R_{\text{full}} = [F_A,\, F_B,\, F_B - F_A,\, |F_B - F_A|,\, F_A \odot F_B]\,.$$

The five components preserve the individual views, their signed change, unsigned discrepancy, and element-wise agreement.

The diagnostic probe uses the absdiff representation, whereas RotationCue uses all five components to recover the structured state. Pair construction is fixed and introduces no learned parameters before the decoder.

F.3 Decoder Architecture

Let d_m denote the Vision L8 feature dimension of base model m. Each component of R_{full} is projected into a shared 512-dimensional latent space,

$$h_j = P_j(r_j), \qquad h_j \in \mathbb{R}^{512},$$

using a Linear–GELU–Dropout projection with normalization as appropriate. Learned component-type embeddings distinguish the five relation components.

The decoder contains three learnable pose queries,

$$Q^{(0)} = [q_s, q_r, q_c],$$

corresponding to same-object, rotation-presence, and angle-range prediction. Two pre-layer-normalized extractor blocks update the queries,

$$Q^{(\ell+1)} = \text{PoseBlock}\big(Q^{(\ell)}, H\big), \qquad \ell \in \{0, 1\}.$$

Each block applies cross-attention from the state queries to the relation tokens, self-attention among the queries, and a $512 \to 2048 \to 512$ feed-forward network. Attention uses eight heads with dropout 0.1.

The resulting query states are mapped to the three predictions,

$$s = g_s(q_s), \qquad r = g_r(q_r), \qquad c = g_c(q_c),$$

where each classifier uses

$$\mathrm{LayerNorm} \to \mathrm{Linear}(512, 512) \to \mathrm{GELU} \to \mathrm{Dropout} \to \mathrm{Linear}\,.$$

Separate state queries preserve the semantic decomposition of the hierarchical relation rather than reducing it to a single flat class.

F.4 Hierarchical Supervision

Let y_s, y_r, and y_c denote the ground-truth same-object, rotation-presence, and angle-range labels. We define

$$m_r = \mathbb{1}[y_s = \texttt{yes}], \qquad m_c = \mathbb{1}[y_s = \texttt{yes} \ \wedge \ y_r = \texttt{yes}],$$

and optimize

$$\mathcal{L}_{\mathrm{base}} = \mathrm{CE}(s, y_s) + \mathrm{CE}_{m_r}(r, y_r) + \mathrm{CE}_{m_c}(c, y_c).$$

The masks prevent supervision of undefined states:

$$y_s = \texttt{no} \Rightarrow y_r, y_c \text{ undefined},$$

and

$$y_r = \texttt{no} \Rightarrow y_c \text{ undefined}.$$

We additionally use an expected angular-cost term,

$$\mathcal{L} = \mathcal{L}_{\mathrm{base}} + \lambda_{\mathrm{angle}} \mathcal{L}_{\mathrm{angle}},$$

where

$$\mathcal{L}_{\mathrm{angle}} = \frac{1}{\sum_i m_{c,i}} \sum_i m_{c,i} \sum_{b=1}^{3} p_{i,b} \min\left(2, \frac{|\theta_i - a_b|}{45}\right),$$

$p_{i,b}$ is the predicted probability of angle range b, and a_b is its predefined angular reference; angles in this expression are measured in degrees. We use

$$\lambda_{\mathrm{angle}} = 0.3.$$

Ordinary cross-entropy treats all incorrect ranges equally. The angular term instead assigns a distance-dependent, capped cost to the predicted ranges, preserving the ordered geometry of the three ranges.

F.5 Training Data and Object-Disjoint Splits

The decoder is trained exclusively on independent multi-view data: ALOI-View (Geusebroek et al., 2005) and SelectiveMV ModelNet40v1 (Alzahrani et al., 2024; Wu et al., 2015).

ALOI contains 1,000 objects observed at 72 azimuths,

$$0^\circ, 5^\circ, \ldots, 355^\circ.$$

For two views with azimuths a and b, the unsigned rotation magnitude is

$$\theta = \min\left(|a - b|, 360^\circ - |a - b|\right).$$

SelectiveMV contains 3,983 objects with 12 RGB views. Each view is assigned

$$a = 30^\circ \times \texttt{view_id}.$$

Splits are constructed at the object level rather than the pair level,

$$\mathcal{O}_{\mathrm{train}} \cap \mathcal{O}_{\mathrm{val}} = \varnothing,$$

preventing different view pairs of the same physical object from entering both partitions.

OR-Bench is used only for final evaluation. Its objects, images, labels, questions, answer choices, and answer formats are not used for decoder training or checkpoint selection.

F.6 Frozen Feature Caching and Shared Training Protocol

Vision L8 representations are extracted once and cached. The decoder is trained on the cached feature pairs while the base VLM remains frozen.

For all three VLMs, the decoder input is the mean-pooled Vision L8 representation. Feature extraction follows the native image interface of each vision backbone, while the decoder design and optimization protocol are kept fixed.

Table 10: Frozen internal representations used as RotationCue inputs.

Base VLM	**Decoder input**	**Image preprocessing**
Qwen3-VL-8B	Vision L8 mean	Qwen AutoProcessor, dynamic resolution
Qwen2.5-VL-7B	Vision L8 mean	Qwen AutoProcessor, dynamic resolution
InternVL3.5-8B	Vision L8 mean	448×448, at most one tile

The two images are encoded independently in all cases, and no language-model hidden state is used as a decoder input.

Across all three VLMs, we use the same readout layer, relation representation, decoder design, and training protocol. A single decoder per base model serves all eight OR-Bench tasks. The shared configuration is summarized below.

Table 11: Shared RotationCue decoder training configuration.

Configuration	**Value**
Training pairs	1,200,000
Validation pairs	50,000
Epochs	6
Batch size	1024
Optimizer	AdamW
Learning rate	10^{-4}
Weight decay	0.01
Gradient clipping	1.0
Training dtype	bfloat16
Seed	20260603
Shuffle	Yes
LR scheduler	None
Relation mode	`rich_pair`
Coarse classification	Hard cross-entropy
Ordinal head	None
Base VLM update	None

Internal validation and checkpoint selection. The training pipeline uses the same fixed internal validation procedure for all three decoders. Table 12 reports the selected checkpoint and its hierarchical-state accuracy on the independent 50,000-pair validation set.

Same-object accuracy is measured on all 50,000 pairs. Rotation-present accuracy is evaluated on the 37,500 same-object pairs for which that state is defined, while angle-range accuracy is evaluated on the 32,500 pairs with an actual rotation. Joint hierarchical accuracy requires all active fields of the final state to be correct.

Across the three models, the two gating states achieve 97.26%–98.70% accuracy, and angle-range accuracy is 63.85%–67.65%.

Table 12: Internal validation performance (%) of the selected RotationCue decoder checkpoints.

Model	Epoch	Same-object	Rotation-present	Angle-range	Joint
Qwen3-VL-8B	5	**98.70**	**98.68**	67.44	**77.75**
Qwen2.5-VL-7B	6	98.03	98.01	63.85	74.95
InternVL3.5-8B	5	97.28	97.26	**67.65**	76.47

F.7 RotationCue for Multi-Image Tasks

RotationCue itself remains strictly pairwise. For a multi-image task, we first construct a task-specific set of image pairs

$$\mathcal{P} = \{(I_{i_k}, I_{j_k})\}_{k=1}^{K},$$

and independently predict one rotation state for each pair,

$$\hat{z}_k = D_\phi\big(R_{\text{full}}(F_{i_k}, F_{j_k})\big), \qquad k = 1, \ldots, K.$$

Each serialized pairwise state forms a context card. The cards are concatenated as structured context,

$$C_{\text{multi}} = C(\hat{z}_1) \oplus \cdots \oplus C(\hat{z}_K),$$

and supplied together with all original images to the frozen VLM,

$$\hat{y} = M_\theta\big(\mathcal{I}, q, C_{\text{multi}}\big).$$

The frozen VLM receives all original images and the pairwise state cards and generates the final answer in a single inference step. No voting or programmatic aggregation is applied to the pairwise predictions.

Table 13: Pair construction for multi-image OR-Bench tasks.

Task	**Constructed pairs**	**# Cards**
View Matching	Ref–A, Ref–B, Ref–C	3
Post-Rotation View	Ref–A, Ref–B, Ref–C, Ref–D	4
Rotation Comparison	A1–A2, B1–B2	2

F.8 Structured State Injection and Controls

At inference time, the decoder predicts

$$\hat{z} = (\hat{z}_s, \hat{z}_r, \hat{z}_c),$$

which is serialized using a fixed field order:

```
same_object: {yes|no}
rotation_present: {yes|no|not_applicable}
angle_range: {small|medium|large|not_applicable}
```

If `same_object=no`, the remaining fields are set to `not_applicable`. If `same_object=yes` and `rotation_present=no`, only `angle_range` is set to `not_applicable`. An angle range is provided only when both gates are positive.

The serialized state is inserted before the task question. The frozen VLM uses this structured context together with the original images for subsequent reasoning.

The complete inference path is

$$(I_A, I_B) \to (F_A, F_B) \to \hat{z} \to C(\hat{z}) \to M_\theta\big(I_A, I_B, q, C(\hat{z})\big).$$

The main-text context-source comparison includes Raw, zero-shot CoT, 5-shot CoT, Self-gen, and RotationCue. Self-gen prompts the base model to generate the same three-field context before producing its final answer, holding the field structure fixed while changing the source of its content.

We use three controlled variants to examine the role of state quality.

Shuffled RotationCue. Predicted states are deterministically permuted across evaluation examples,

$$x_i \to C(\hat{z}_{\pi(i)}),$$

breaking state–input correspondence while preserving the same state format and marginal text distribution.

RotationCue. Each example receives the state recovered from its own frozen representations,

$$x_i \to C(\hat{z}_i).$$

Oracle RotationCue. The same interface is populated with the metadata-derived ground-truth state,

$$x_i \to C(z_i^\star).$$

Oracle RotationCue provides a ground-truth-state reference. The aggregate state-quality results are reported in the main text; across all three models they follow

$$\text{Shuffled} < \text{Raw} < \text{RotationCue} < \text{Oracle}.$$

Because Shuffled and Oracle retain the same structured interface while altering only state–input correspondence or state correctness, these controls distinguish useful state recovery from a generic benefit of adding rotation-related text.

F.9 RotationCue Context Field Ablation

We ablate the fields included in the RotationCue context to examine their contributions. We compare the full context with two partial variants: *Same + Rotation*, which retains only the same-object and rotation-presence fields, and *Angle Range*, which retains only the predicted coarse angle range.

Table 14: Ablation of the fields included in the RotationCue context. Values are Macro-8 scores (%); values in parentheses are gains over Raw (Δ, percentage points).

Base VLM	Raw	Same + Rotation	Angle Range	Full RotationCue
Qwen3-VL-8B	46.8	48.5 (+1.7)	56.4 (+9.6)	**58.0** (+11.2)
Qwen2.5-VL-7B	42.3	43.4 (+1.1)	50.7 (+8.4)	**54.9** (+12.6)
InternVL3.5-8B	43.0	45.6 (+2.6)	50.3 (+7.3)	**50.9** (+7.9)

Across all three VLMs, the angle-range field provides most of the improvement over Raw, whereas the same-object and rotation-presence fields alone yield only modest gains. The full RotationCue context remains strongest, indicating that the additional fields provide complementary information.

F.10 Efficiency and Stochastic-Decoding Robustness

Under the same inference setup, RotationCue processes an example in approximately 0.9 seconds on average, compared with 2.2 seconds for zero-shot CoT, and adds only 1.2% latency over Raw. This corresponds to approximately $2.4\times$ faster inference than zero-shot CoT. Decoder training requires 0.23 A100 GPU-hours in the Qwen3-VL-8B comparison reported in the main text.

We further evaluate robustness when the final answer is sampled rather than greedily decoded. On Qwen3-VL-8B, 400 examples are independently decoded nine times using

$$T = 0.5, \qquad \text{top-p} = 0.95.$$

Accuracy standard deviation is the population standard deviation across the nine run-level accuracies. Exact Consistency measures the fraction of examples for which all nine sampled answers are identical,

$$\text{EC} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\hat{y}_i^{(1)} = \cdots = \hat{y}_i^{(9)}\right],$$

and

$$\mathrm{FlipRate} = 1 - \mathrm{EC}\,.$$

We additionally report Correct-Stable, the fraction of examples for which all nine sampled answers equal the ground-truth answer,

$$\mathrm{CS} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\hat{y}_i^{(1)} = \cdots = \hat{y}_i^{(9)} = y_i\right].$$

Prediction entropy summarizes the dispersion of the nine sampled answers for each example; lower values indicate a more concentrated prediction distribution.

Table 15: Accuracy and prediction stability under stochastic decoding on Qwen3-VL-8B. Each of 400 examples is decoded nine times with temperature 0.5 and top-$p = 0.95$.

Method	Acc. Mean	Acc. Std	Exact Cons.	Flip Rate	Entropy	Correct-Stable
Raw	56.6	0.7	84.0	16.0	0.131	50.5
Zero-shot CoT	58.2	1.2	71.3	28.8	0.263	47.3
RotationCue	**64.8**	**0.5**	**86.0**	**14.0**	**0.111**	**59.0**

RotationCue retains its accuracy advantage under stochastic decoding while producing more consistent predictions than zero-shot CoT. Its Exact Consistency reaches 86.0%, compared with 71.3% for zero-shot CoT, while Correct-Stable performance increases from 47.3% to 59.0%.

G External Spatial Evaluation

We additionally evaluate the RotationCue inference pipeline on VSR-ZS, CV-Bench, and SpinBench to assess whether the structured-state interface preserves performance on broader spatial reasoning benchmarks. We use 1,222 examples from VSR-ZS, 2,638 from CV-Bench, and the 2,739-example SpinBench test split.

G.1 Input Construction

RotationCue always operates on image pairs. For the single-image VSR-ZS and CV-Bench tasks, we duplicate the input image and construct

$$(I_A, I_B) = (I, I),$$

so that the original visual content remains unchanged while retaining the pairwise RotationCue interface. The resulting structured state is provided together with the original task input to the frozen VLM. This construction evaluates whether introducing the structured-state interface preserves performance on single-image spatial reasoning tasks.

SpinBench contains heterogeneous single- and multi-image tasks. We therefore use a task-aware adapter that constructs pairwise RotationCue inputs when the task defines a meaningful object relation. For reference-to-candidate tasks, the adapter forms

$$\mathcal{P} = \{(\mathrm{Ref}, C_1), \ldots, (\mathrm{Ref}, C_K)\},$$

while explicit two-view rotation tasks use their provided image pair. Identity or odd-one-out tasks may use all pairwise combinations,

$$|\mathcal{P}| = \binom{N}{2}.$$

Each pair is processed independently by the same RotationCue decoder,

$$\hat{z}_k = D_\phi\big(R_{\mathrm{full}}(F_{i_k}, F_{j_k})\big),$$

and all resulting cards are supplied together with the complete original image set and question. The frozen VLM generates the final answer in a single inference step.

The angle field represents unsigned rotation magnitude; task-specific interpretation remains with the VLM.

For tasks without a meaningful pairwise rotation relation, we retain the original Raw prompting. This routing is determined solely by the task structure, independently of ground-truth labels, model predictions, or task performance. Under this fixed policy, RotationCue is invoked for 497 examples and Raw inference is retained for the remaining 2,242 examples. We report the full-test score of the resulting task-aware pipeline.

Table 16: External spatial benchmarks and RotationCue input construction.

Benchmark	# Examples	RotationCue construction
VSR-ZS	1,222	Duplicate single image: I–I
CV-Bench	2,638	Duplicate single image: I–I
SpinBench	2,739	Task-aware pair construction with Raw fallback

H Rotation-SFT: Data Construction and Training Details

Rotation-SFT provides a supervision-aligned alternative to RotationCue. Both methods use the same external multi-view sources, rotation semantics, and angular intervals, but apply the supervision differently. RotationCue trains an external decoder over frozen internal representations, whereas Rotation-SFT directly adapts the language side of Qwen3-VL-8B through LoRA. The vision encoder and multimodal projector remain frozen.

H.1 External Training Data and Supervision Alignment

Rotation-SFT is constructed from the same independent multi-view sources used for RotationCue, namely ALOI-View (Geusebroek et al., 2005) and SelectiveMV ModelNet40v1 (Alzahrani et al., 2024; Wu et al., 2015). We retain the same rotation definitions and coarse angular intervals, so that the supervision semantics are aligned between the two methods.

The full SFT training set contains 900,000 examples. A separate class-balanced validation set contains 20,000 examples and is used for checkpoint selection.

OR-Bench examples are not used to construct the SFT training or validation data.

H.2 Conversion to Four-Way VQA

RotationCue represents the relation hierarchically through `same_object`, `rotation_present`, and `angle_range`. For Rotation-SFT, same-object pairs are rewritten as two-image, four-way VQA examples.

The training prompt uses the following fixed prefix:

```
<image><image>
Image A is shown first and Image B second.
Both images show the same object.
Which option best describes the relative rotation
from Image A to Image B?
Answer with the letter only.
```

The answer options correspond to four semantic classes. For a pair with unsigned rotation magnitude θ, the semantic target is

$$y_{\mathrm{SFT}}(\theta) = \begin{cases} \texttt{no_rotation}, & \theta = 0^\circ, \\ \texttt{small}, & 0^\circ < \theta < 60^\circ, \\ \texttt{medium}, & 60^\circ \le \theta \le 120^\circ, \\ \texttt{large}, & 120^\circ < \theta \le 180^\circ. \end{cases}$$

Thus, Rotation-SFT preserves RotationCue's zero-rotation and angular-range semantics for same-object pairs in a four-way VQA target.

H.3 Answer-Option Construction

We avoid a fixed correspondence between semantic classes and answer letters. The correct answer position is cycled deterministically across classes, while the ordering of the remaining choices is determined from a SHA-256 hash of the source sample identifier. Consequently,

$$y_{\text{semantic}} \not\equiv y_{\text{letter}},$$

preventing the model from solving the training task through a stable class-to-letter shortcut.

H.4 Language-Side LoRA Adaptation

Rotation-SFT adapts Qwen3-VL-8B using LoRA on the language side. The vision encoder and multimodal projector remain frozen throughout training. The LoRA configuration is

$$r = 8, \qquad \alpha = 16, \qquad p_{\text{drop}} = 0.05,$$

with the target setting `all`.

Accordingly, visual feature extraction and multimodal projection remain unchanged, while the language-side LoRA parameters receive the rotation VQA supervision. This differs from RotationCue, for which the entire base VLM remains frozen and only the external decoder is trained.

H.5 Training Configuration

The reported full-scale run uses all 900,000 training examples for one epoch. Table 17 summarizes the training configuration.

Table 17: Rotation-SFT training configuration.

Configuration	Value
Base model	Qwen3-VL-8B
Training examples	900,000
Validation examples	20,000
Epochs	1
LoRA rank	8
LoRA alpha	16
LoRA dropout	0.05
LoRA target setting	`all`
Per-device batch size	4
Gradient accumulation	4
Learning rate	10^{-4}
LR scheduler	Cosine
Warmup ratio	3%
Training dtype	bfloat16
Vision encoder	Frozen
Multimodal projector	Frozen
Distributed processes	2
Hardware	2× A100
Training time	$\sim$ 21 hours
Training cost	$\sim$ 42 A100 GPU-hours

H.6 External Validation and Checkpoint Selection

Checkpoint selection is performed exclusively on the independent 20,000-example class-balanced validation set from the external multi-view training sources. For a candidate checkpoint c, we compute four-way VQA accuracy,

$$\text{Acc}_{\text{val}}(c) = \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \mathbb{1}\left[\hat{y}_i^{(c)} = y_i\right], \qquad N_{\text{val}} = 20{,}000.$$

The reported checkpoint is fixed as

$$c^\star = \arg\max_c \mathrm{Acc}_{\mathrm{val}}(c).$$

OR-Bench is not used for optimization or checkpoint selection and is evaluated only after $c^\star$ has been fixed. The resulting protocol is

external training $\rightarrow$ external validation $\rightarrow$ checkpoint selection $\rightarrow$ OR-Bench evaluation.

H.7 Task-Level Results

For completeness, Table 18 reports the complete OR-Bench task-level results of the selected Rotation-SFT checkpoint. All deltas are computed from displayed scores and reported in percentage points.

Table 18: Task-level accuracy (%) of Qwen3-VL-8B before and after Rotation-SFT. Macro-8 gives equal weight to the eight tasks, whereas Micro-8 aggregates all 2,109 examples. Deltas are computed from displayed scores.

Task	**N**	**Raw**	**Rotation-SFT**	**Δ**
Rotation Detection	486	**99.6**	89.9	−9.7
Coarse Angle	243	40.7	**85.2**	+44.5
90° Threshold	243	37.0	**41.6**	+4.6
Exact Angle (MCQ)	243	24.7	**40.7**	+16.0
Exact Angle (Open)	243	25.5	**27.6**	+2.1
View Matching	216	**65.3**	54.2	−11.1
Rotation Comparison	243	48.6	**49.8**	+1.2
Post-Rotation View	192	**32.8**	20.3	−12.5
Macro-8	—	46.8	**51.2**	+4.4
Micro-8	2,109	53.0	**56.3**	+3.3