Readable Yet Unpredictable: Rotated-Outcome Prediction in Vision-Language Models

Lexin Wang[1,2,3], Shenghua Liu[1,2,3]*, Yiwei Wang[4], Jiafeng Guo[1,2,3], Xueqi Cheng[1,2,3]

[1]State Key Laboratory of AI Safety
[2]Institute of Computing Technology, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]University of California, Merced
https://rotoutbench.netlify.app/
lizzie011@163.com, liushenghua@ict.ac.cn

Abstract

Can vision-language models predict what a 180° rotation would reveal from the original image alone? We study this ability through *Rotated-Outcome Prediction*: given an original image, a model must answer what would be seen or read after a 180° in-plane rotation, without directly observing the rotated target. To study this gap, we introduce ROTOUTBENCH, a paired diagnostic benchmark spanning open visual cases and controlled text-image rotations. A sharp pattern emerges: many VLMs can recognize the relevant content when directly given either the original or rotated image, yet fail to infer the rotated result from the original image alone. On controlled text-image rotations, predicted-rotation accuracy collapses to near zero even for models with high direct-reading accuracy. A model-level diagnostic further shows that the prediction state can approach a rotated-image reading state, while the final readout still shifts toward the original string. Current VLMs can recognize a transformed visual state when it is shown, but often fail to predict that state from the original view.

1 Introduction

Vision-language models (VLMs) (Radford et al., 2021; Li et al., 2022; Alayrac et al., 2022; Liu et al., 2023b) have achieved strong performance on image understanding and multimodal reasoning tasks. Yet visual inputs are not always observed in a canonical orientation: documents may be scanned upside down, text may appear at unusual angles, and agents may view objects from non-standard viewpoints. In such cases, recognizing the current image is not always enough; the model may need to infer how the content would appear after a change of orientation.

Existing benchmarks study important parts of this problem. OCR and document-understanding benchmarks evaluate whether models can read visible text (Liu et al., 2024b; Fu et al., 2025b), while spatial, orientation, and rotation benchmarks test whether models can judge relations, viewpoints, or rotated inputs that are directly presented (Stogiannidis et al., 2025; Jung et al., 2025; Niu et al., 2026). These evaluations show that orientation and spatial structure remain challenging for VLMs. However, they mainly evaluate recognition or judgment over visual states that are already shown to the model.

This leaves an under-explored distinction: recognizing a transformed visual state when it is shown is not the same as predicting that state from an observed starting point. We study this ability through *Rotated-Outcome Prediction*. Given an original image, the model must answer what would be seen or read after a 180° in-plane rotation, without directly observing the rotated target. This setting tests whether the model can infer an unseen rotated outcome from the original image alone.

To study this ability, we introduce ROTOUTBENCH, a paired diagnostic benchmark with two complementary subsets. Visual-Rot contains 68 paired examples, yielding 136 image views, where natural and icon-like images may change interpretation after rotation. TextImage-Rot contains 342 paired string examples, yielding 684 image views, where each sample has an original rendered string image and a canonical rotated target image. ROTOUTBENCH is designed as a focused diagnostic benchmark rather than a large-scale leaderboard: its paired design prioritizes controlled comparison between endpoint recognition and unseen outcome prediction. Figure 1 shows representative examples

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 22095–22111
October 24-29, 2026 ©2026 Association for Computational Linguistics

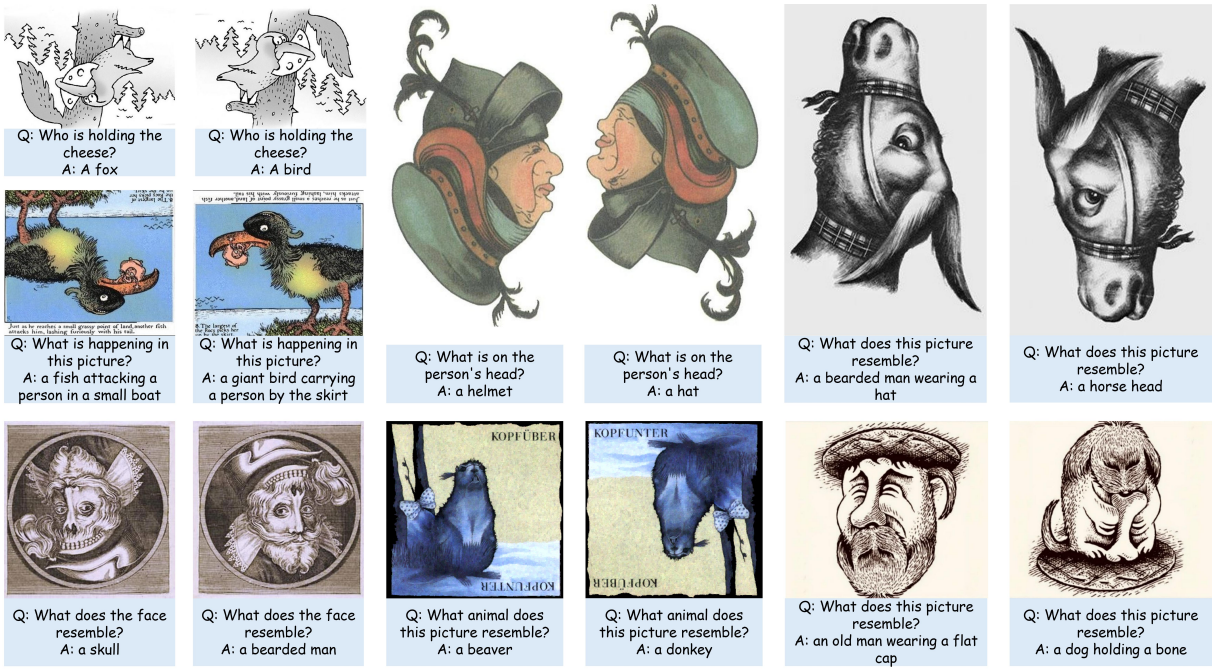

Figure 1: Examples from Visual-Rot. The benchmark includes images whose interpretation can change under a 180° rotation, requiring models to predict an unobserved rotated visual outcome rather than only recognize the currently visible image.

from the Visual-Rot subset.

A striking pattern emerges on TextImage-Rot: many recognition-capable VLMs can directly read the relevant content when given either the original or rotated image, yet their predicted-rotation accuracy collapses to near zero when asked to infer the rotated result from the original image alone. This is not simply a failure to recognize rotated images. Instead, it reveals an endpoint-recognition vs. outcome-prediction disconnect: current VLMs can recognize visible transformed states, but do not reliably derive those states from observed inputs.

We further probe this failure with text-only controls and model-level diagnostics. The analysis suggests that the prediction state can approach a rotated-image reading state, but the final readout still favors the original string. Across several models, patching a state from direct rotated-image reading shifts the readout toward the rotated answer.

Our contributions are threefold:

- We formalize *Rotated-Outcome Prediction*, a diagnostic setting for predicting an unseen rotated outcome from an observed image.
- We introduce ROTOUTBENCH, a paired benchmark with open visual cases and controlled text-image rotations.
- We show that VLMs can recognize transformed endpoints when shown, yet fail to predict the rotated outcome from the original view.

2 Related Work

2.1 Orientation and Spatial Reasoning in VLMs

Spatial reasoning has long been used to diagnose visual understanding, from compositional datasets such as CLEVR (Johnson et al., 2017) and GQA (Hudson and Manning, 2019) to spatial-relation benchmarks such as VSR (Liu et al., 2023a) and What'sUp (Kamath et al., 2023). Recent diagnostic suites further show that VLMs may perform well on general VQA while still failing on fine-grained spatial, geometric, and viewpoint-sensitive tasks (Tong et al., 2024; Fu et al., 2024; Li et al., 2024; Wang et al., 2024; Stogiannidis et al., 2025). Visual text and document-understanding benchmarks further show that reading visible text remains an important capability for multimodal models (Liu et al., 2024b; Fu et al., 2025b). These failures suggest that VLMs do not always represent visual text, relations, orientation, and viewpoint-dependent structure robustly.

More closely related are benchmarks on orientation, reference frames, and mental rota-

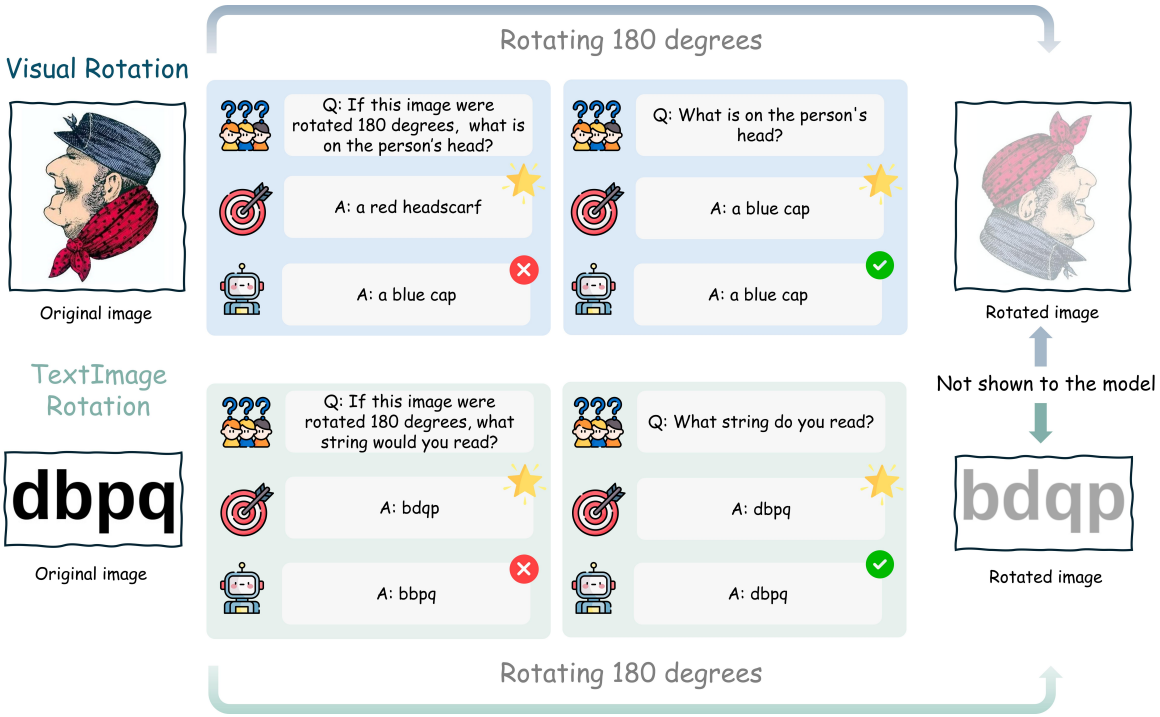

Figure 2: Example cases from Visual-Rot and TextImage-Rot. Each example pairs an original view with its canonical 180° rotated view. The examples illustrate the gap between directly reading a visible rotated target and predicting the same rotated outcome from the original view.

tion. EgoOrientBench (Jung et al., 2025) and perspective-aware reasoning tasks (Lee et al., 2025b; Zhang et al., 2025b) show that models can struggle with egocentric directions and viewpoint changes. RotBench (Niu et al., 2026) evaluates whether MLLMs can identify image orientations, while SpinBench (Zhang et al., 2025b) studies rotation and perspective taking as spatial-reasoning probes. These works test orientation recognition, comparison, or perspective reasoning when the relevant visual inputs or candidate views are provided. Our setting differs in that the rotated image is not shown: the model must predict the rotated visual-semantic outcome from the original image.

2.2 Visual Grounding, Ambiguity, and Language Priors

Another related line of work studies when VLM outputs are not fully grounded in the current visual evidence. Object hallucination and language-prior benchmarks show that models may produce answers driven by dataset priors or language bias rather than the image (Rohrbach et al., 2018; Li et al., 2023; Lee et al., 2025a; Vo et al., 2025). Broader evaluation suites such as MME (Fu et al., 2025a) also expose perception and cognition failures not captured by standard VQA metrics.

Visual illusion and grounding benchmarks study difficult, ambiguous, or misleading images that can affect model judgments (Guan et al., 2024; Zhang et al., 2023; Shahgir et al., 2024; Zhang et al., 2025a). Our setting differs in that the target visual state is not shown: the model must infer a rotated outcome from the original image. Related work on visual anagrams, ambigram generation, and multi-view optical illusions shows that visual meaning can change with viewing orientation (Shirakawa and Uchida, 2023; Geng et al., 2024). These works primarily focus on generating orientation-dependent images, whereas our goal is to evaluate whether VLMs can infer an unobserved rotated outcome from the original image.

3 Methodology

3.1 Task Definition

We study *Rotated-Outcome Prediction*: whether VLMs can infer what would be seen or read after a 180° in-plane rotation without directly observing the rotated target. Given an original image I, we define $R_{180}(I)$ as the image obtained by rotating I by 180° within the image plane. The model is shown only I and is asked to answer what would be seen or read in the rotated view $R_{180}(I)$.

This task differs from standard rotation robust-

ness evaluation, where the rotated image is directly provided to the model. In our setting, the model must infer the answer associated with an unobserved rotated visual state. This allows us to separate direct recognition of a rotated input from prediction of the rotated outcome based on the original image.

We focus on 180° rotation because it often yields stable, verifiable semantic changes in visual text and orientation-sensitive images, allowing us to compare endpoint recognition with unseen outcome prediction in a controlled setting.

3.2 ROTOUTBENCH Construction

We construct ROTOUTBENCH, a paired diagnostic benchmark with two subsets: Visual-Rot and TextImage-Rot. Each base example is a rotation pair consisting of an original view and its canonical 180° rotated view. This paired design supports three conditions on the same underlying example: direct reading of the original view, direct reading of the rotated view, and prediction of the rotated target from the original view.

Visual-Rot is built from publicly available web images whose interpretation may change after a 180° rotation. Candidate images include inversion-dependent illustrations, ambigram-like graphics, directional icons, and dual-interpretation images. We manually filter candidates to remove low-quality, overly ambiguous, visually unsupported, or answer-leaking cases. Each retained example is reviewed by two annotators to ensure that the original view, rotated view, question, reference answer, and acceptable keywords are mutually consistent. Disagreements or uncertain cases are resolved by discussion, and examples without a stable rotated interpretation are removed. This process yields 68 paired examples, or 136 image views. We also construct 68 four-way rotation matching questions from these paired examples.

Because Visual-Rot uses open-ended questions, we use manually verified acceptable keywords to reduce evaluation ambiguity. These keywords cover reasonable synonyms, visually equivalent names, and specific category-level descriptions, while excluding overly broad or visually unsupported answers. Thus, Visual-Rot is intended as an open visual diagnostic rather than a fully automatic large-scale leaderboard.

TextImage-Rot provides a more controlled counterpart. It contains 342 paired string examples, yielding 684 image views. Each example has an original rendered string image and a canonical rotated target image. The reference answer is produced by a deterministic 180° rotation rule with character mapping and sequence reversal. Characters without well-defined rotated readings are excluded. The current split balances strings of length 1–5: all single-character and two-character combinations are included, while longer strings are balanced-sampled to avoid the evaluation being dominated by combinatorially many long strings. This keeps the split compact, controlled, and extensible to more characters, fonts, styles, and string lengths.

Together, the two subsets trade off openness and control. Visual-Rot tests whether the phenomenon appears in natural and icon-like images, while TextImage-Rot isolates the recognition-versus-prediction distinction under programmatically generated references. Figure 2 illustrates the evaluation format; additional construction and evaluation details are provided in Appendix A.

3.3 Metrics and Evaluation Protocol

All reported scores are empirical accuracies over the paired examples in each subset. We denote model outputs as $\hat{Y}_X^Q$, where X specifies the input shown to the model and Q specifies the question type. We use I_{orig} for the original image, I_{rot} for the rotated target image, Q_d for direct-reading questions, and Q_p for predicted-rotation questions. Using this notation, we define three main scores:

$$R_{\text{orig}} = \text{Acc}(\hat{Y}_{I_{\text{orig}}}^{Q_d}, Y_{\text{orig}}), \quad (1)$$

$$R_{\text{rot}} = \text{Acc}(\hat{Y}_{I_{\text{rot}}}^{Q_d}, Y_{\text{rot}}), \quad (2)$$

$$R_{\text{pred}} = \text{Acc}(\hat{Y}_{I_{\text{orig}}}^{Q_p}, Y_{\text{rot}}). \quad (3)$$

Exact prompt templates and the image-specific Visual-Rot question variants are provided in Appendix A.

We also report $\text{Gap} = R_{\text{rot}} - R_{\text{pred}}$, which measures how much easier it is to recognize the rotated target when shown than to infer it from the original view.

For Visual-Rot and TextImage-Rot, R_{orig}, R_{rot}, and R_{pred} are computed over paired examples. Thus, the main reported scores are computed over 68 Visual-Rot pairs and 342 TextImage-Rot pairs. The auxiliary rotation matching task is evaluated over 68 four-way matching questions constructed from the paired Visual-Rot image views.

For Visual-Rot, open-ended answers are evaluated by normalized keyword matching against

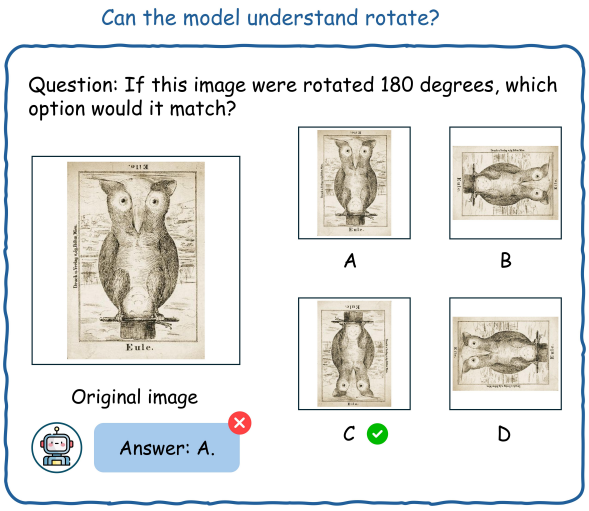

Figure 3: Example of the four-way rotation matching task in Visual-Rot. The model is given a query image and four candidates, and must select the candidate corresponding to its 180° rotated view. We use this task as an auxiliary diagnostic.

manually verified references and acceptable keywords. This evaluation is designed to allow natural answer variation while rejecting overly broad or visually unsupported responses. For TextImage-Rot, outputs are evaluated by normalized string matching against programmatically generated references. Additional details on dataset size, prompt templates, answer parsing, keyword matching, TextImage-Rot generation, and rotation matching are provided in Appendix A.

4 Behavioral Results

4.1 Experimental Setup

We evaluate nine representative VLMs on ROTOUTBENCH: Qwen2.5-VL-3B (Bai et al., 2025b), Qwen3-VL-8B (Bai et al., 2025a), Qwen3.5-9B (Qwen Team, 2026), Gemma3-4B (Gemma Team et al., 2025), Kimi-VL-A3B (Kimi Team et al., 2025), MiniCPM-V-2.6 (Yao et al., 2024), MiniCPM-V-4.5 (Yu et al., 2025), InternVL3.5-8B (Wang et al., 2025), and LLaVA-1.5-7B (Liu et al., 2024a). All models are evaluated with the protocol defined in Section 3. Table 1 reports direct reading, predicted rotation, the resulting gap, and the auxiliary four-way rotation matching score.

4.2 Open Visual Rotation

Open visual cases reveal the prediction challenge. Visual-Rot tests rotation-dependent interpretation in open-ended images whose meaning may change after a 180° rotation. As shown in Table 1, R_{pred} remains low across models, showing that models struggle to infer the rotated outcome from the original image. These results show that rotated-outcome prediction is difficult in open visual cases where rotation changes the apparent object, relation, direction, or semantic interpretation.

Prediction remains difficult even when both views are correctly read. Some models read rotated views less accurately than original views, raising the possibility that low prediction accuracy is partly due to difficulty in recognizing the rotated target itself. To control for this, we evaluate prediction only on pairs for which the same model correctly reads both the original and rotated views. This subset is determined using only the two direct-reading results, without using the prediction output. As shown by R_{dual} in Table 1, prediction remains low for most models.

Showing candidate images still does not make the correspondence reliable. The four-way rotation matching task asks models to select the 180° rotated counterpart from candidate images, as illustrated in Figure 3. Results remain near or below the 25% random baseline. This provides complementary evidence that the rotation correspondence is difficult even when candidate views are explicitly shown. Since this task evaluates candidate selection rather than open-ended rotated-outcome prediction, we report it as an auxiliary diagnostic alongside the main reading and prediction scores.

4.3 Controlled Text-Image Rotation

TextImage-Rot provides a controlled endpoint comparison. TextImage-Rot complements the open visual cases with a controlled setting where each sample has an original string image, a canonical rotated target image, and a programmatically generated rotated answer.

Readable endpoints still lead to near-zero prediction. Table 1 shows a sharp separation between direct reading and predicted rotation. For models with strong direct-reading accuracy, both R_{orig} and R_{rot} are high, confirming that the original string and the rotated target are directly readable. However, R_{pred} remains near zero, with the best model reaching only 6.14%.

Simple prompt variants do not remove the failure. We rerun the TextImage-Rot predicted-rotation condition with prompt variants that clarify

Model	Visual-Rot					Rotation Matching		TextImage-Rot			
	R_{orig}	R_{rot}	R_{pred}	R_{dual}	Gap	Acc.	Δ_{rand}	R_{orig}	R_{rot}	R_{pred}	Gap
Qwen2.5-VL-3B	55.88	41.18	8.82	10.00	32.36	23.53	-1.47	**99.42**	**99.42**	0.29	99.12
Qwen3-VL-8B	72.06	**58.82**	17.65	9.38	41.17	25.00	0.00	93.27	94.74	2.05	92.69
Qwen3.5-9B	73.53	57.35	25.00	**40.74**	32.35	25.00	0.00	96.20	96.20	2.34	93.86
Gemma3-4B	55.88	42.65	14.71	31.58	27.94	20.59	-4.41	23.68	23.98	0.29	23.68
Kimi-VL-A3B	57.35	45.59	14.71	21.05	30.88	25.00	0.00	98.83	99.12	2.05	97.08
MiniCPM-V-2.6	63.24	44.12	10.29	10.53	33.83	23.53	-1.47	97.95	97.66	0.58	97.08
MiniCPM-V-4.5	69.12	57.35	**26.47**	34.48	30.88	**26.47**	1.47	95.32	96.49	**6.14**	90.35
InternVL3.5-8B	66.18	47.06	14.71	22.73	32.35	17.65	-7.35	89.18	89.77	2.05	87.72
LLaVA-1.5-7B	**75.00**	57.35	25.00	32.26	32.35	25.00	0.00	13.16	12.57	1.17	11.40

Table 1: Behavioral results on ROTOUTBENCH. All values are percentages. R_{orig} and R_{rot} measure direct recognition of the original and rotated views, while R_{pred} measures prediction of the rotated answer from the original view. R_{dual} measures prediction accuracy on the subset of pairs for which the same model correctly reads both the original and rotated views. $\text{Gap} = R_{\text{rot}} - R_{\text{pred}}$ measures the difference between recognizing the rotated target when shown and inferring it from the original view. Main reading and prediction scores are computed over paired examples; Visual-Rot Rotation Matching is evaluated over 68 four-way matching questions constructed from the paired Visual-Rot image views, with Δ_{rand} denoting the difference from the 25% random baseline.

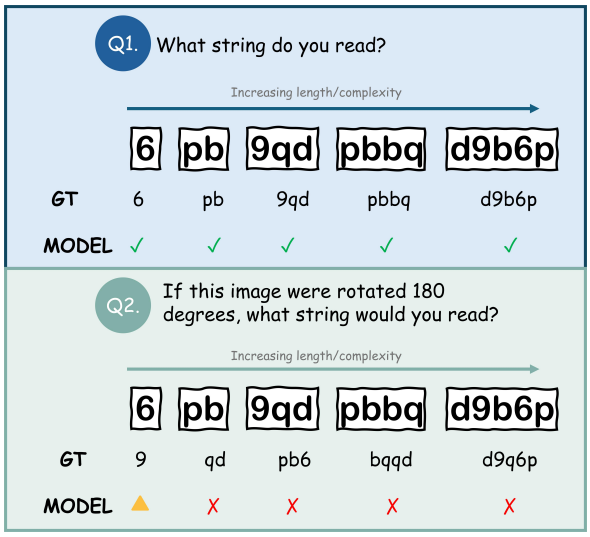

Figure 4: Qualitative example from TextImage-Rot. The model reads the relevant string when directly given either the original or rotated image, but fails to infer the rotated reading from the original image alone. The yellow triangle indicates mixed prediction results across single-character examples.

the input view, discourage copying, make the transformation stepwise, or provide a high-level rotation rule. Predicted-rotation accuracy remains near zero across models, and alternative prompts do not produce a consistent improvement. These results suggest that the failure is not merely an artifact of a single prompt wording. Full prompts and results are provided in Appendix B.2.

Figure 4 shows the same pattern qualitatively: the model reads visible strings correctly, but fails to predict what would be read after a 180° rotation.

High direct reading does not imply prediction. The key pattern is clear among models that can directly read both endpoints. On TextImage-Rot, several models achieve high direct-reading accuracy on both the original and rotated images, yet their predicted-rotation accuracy remains near zero. This rules out a simple endpoint-recognition explanation.

4.4 Text-Only Control

A similar gap appears without image input. For models that support text-only prompts, we use the same underlying strings as TextImage-Rot and ask the model either to directly read the given string or to predict what the string would become after a 180° rotation. This mirrors the visual setting, but replaces image inputs I_{orig} and I_{rot} with text inputs T_{orig} and T_{rot}. As shown in Figure 5, direct reading remains much easier than predicted rotation. Thus, a similar direct-reading versus prediction gap also appears when the input is provided directly as text.

We further test whether the way the input string is presented affects prediction. We consider three formats: a continuous string such as `6bd`, space-separated characters such as `6 b d`, and an explicit character list such as `['6', 'b', 'd']`. The latter two formats make the character boundaries explicit and reduce possible ambiguity in how the string is segmented.

Table 2 reports prediction accuracy under these three input formats. Prediction remains similarly low across all three settings, showing that making the character boundaries explicit does not substan-

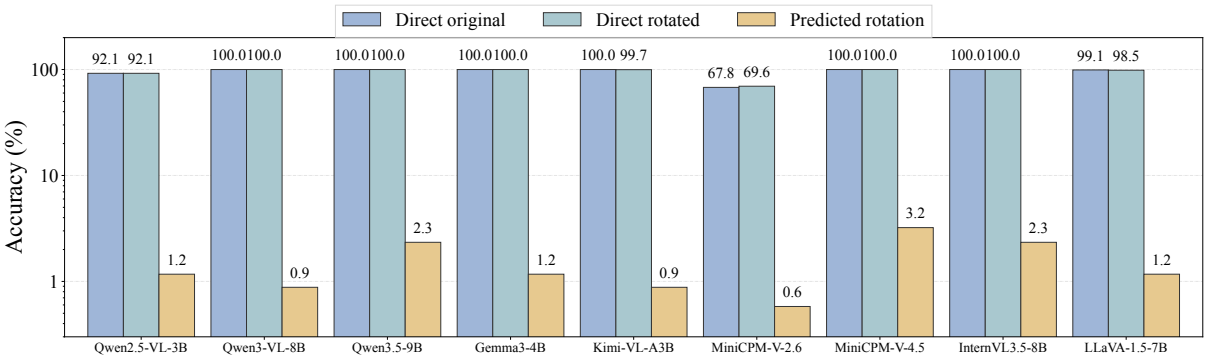

Figure 5: Text-only control results across models that support text-only prompts. Direct reading of the presented strings remains easy for most models, while predicting their rotated forms remains difficult. A similar prediction difficulty therefore also appears without image input.

	Input format		
Model	Continuous	Space-separated	Character list
Qwen2.5-VL-3B	1.2	0.0	0.0
Qwen3-VL-8B	0.9	0.6	1.8
Qwen3.5-9B	2.3	3.5	3.8
Gemma3-4B	1.2	0.6	0.3
Kimi-VL-A3B	0.9	0.9	0.3
MiniCPM-V-2.6	0.6	0.3	0.0
MiniCPM-V-4.5	3.2	3.5	3.5
InternVL3.5-8B	2.3	1.5	2.3
LLaVA-1.5-7B	1.2	1.5	0.0

Table 2: Text-only prediction accuracy (%) under three input formats.

tially improve prediction. Overall, the text-only prediction difficulty persists across different ways of presenting the input string.

5 Internal Diagnostics

We next examine how this behavioral difference appears during model inference. We analyze hidden-state similarity, layer-wise readout, state patching, and attention allocation to better understand how prediction differs from direct rotated-image reading.

5.1 Hidden-State and Readout Analysis

The prediction state resembles rotated-image reading, while the output still favors the original string. We examine whether the image-based prediction state is similar to states observed under the rotated-image and text-only controls. We first compare hidden-state distances across these conditions, and then examine the layer-wise scores assigned to the rotated and original answers.

We compare the image-based prediction state with three diagnostic controls. The first is direct reading of the rotated image, denoted as $h^{Q_d}_{I_{\text{rot}}}$, where the rotated target is visible and the model simply reads it. The second is prompt-matched reading of the rotated image, denoted as $h^{Q_m}_{I_{\text{rot}}}$, where the rotated target is visible but the prompt still contains rotation-related wording. The third is text-only predicted rotation, denoted as $h^{Q_p}_{T_{\text{orig}}}$, where the model receives the original string as text and predicts its rotated form. Full notation is provided in Appendix C.

Figure 6(a) shows the cosine distance from the image-based prediction state to each control across layers. Lower distance means greater hidden-state similarity. The prediction state is closest to the prompt-matched rotated-image control and farther from the text-only prediction control. Thus, it is more similar to the state formed when reading the visible rotated target than to the text-only prediction state. This comparison measures similarity between conditions; it does not show that the model has already formed the correct rotated answer.

We project the layer-wise hidden state to the vocabulary space and compare the logits assigned to the rotated target and the original string. Their difference defines the readout margin. In Figure 6(b), the purple curve uses the first answer token and the yellow curve averages over answer tokens; positive margin favors the rotated target, while negative margin favors the original string. Both readout views shift toward the original string in late layers. Figure 6(c) further reports the fraction of samples whose readout margin is positive, with the dashed line marking a majority threshold. The rotated target is not favored for a stable majority of samples. Together, the hidden-state and readout results show that the prediction state can resemble rotated-image reading while the output still favors

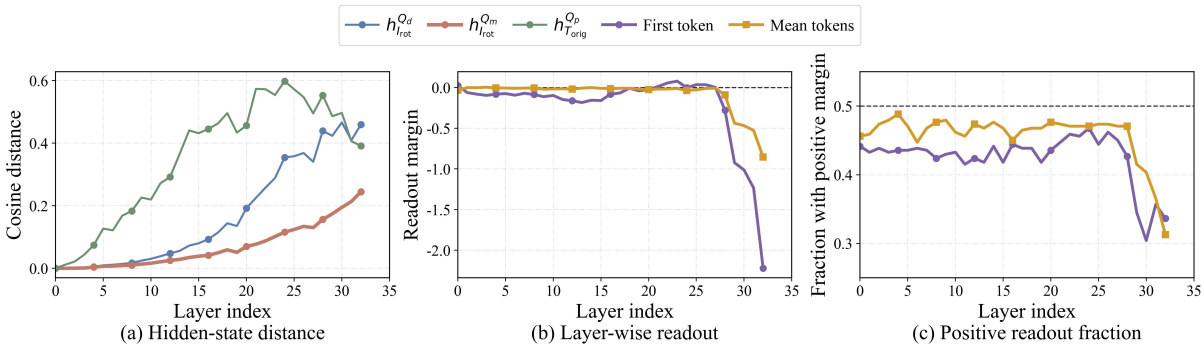

Figure 6: Hidden-state and readout analysis on Qwen3.5-9B. (a) We compare the image-based prediction state with three diagnostic controls: direct rotated-image reading ($h^{Q_d}_{I_{\text{rot}}}$), prompt-matched rotated-image reading ($h^{Q_m}_{I_{\text{rot}}}$), and text-only predicted rotation ($h^{Q_p}_{T_{\text{orig}}}$). Lower distance means greater hidden-state similarity. (b) Layer-wise readout margin between the rotated target and the original string. Positive values favor the rotated target; negative values favor the original string. We report both first-token and mean-token margins. (c) Fraction of samples with positive readout margin; the dashed line marks 0.5. The rotated target is not favored for a stable majority of samples. Detailed notation and computation are provided in Appendix C.

the original string. We next test whether patching a state from direct rotated-image reading changes this preference.

5.2 Patching from Direct Rotated-Image Reading

Patching from direct rotated-image reading shifts the readout toward the rotated answer. We patch the last-input-token state from direct reading of the corresponding rotated image into the predicted-rotation run, using the state from direct reading of the original image as a control. We evaluate pairs whose original and rotated answers differ and whose first answer tokens are different, so that the first-token margin is informative. We evaluate patching across five models: four with Qwen-family language backbones and Phi-4-Multimodal (Abouelenin et al., 2025), which uses a Phi-4-Mini backbone.

Figure 7 shows only a small relative effect in early and middle layers, while patching from the corresponding rotated-image reading produces a stronger shift toward the rotated answer in later layers across all five models. This shows that states elicited by directly viewing the corresponding rotated image can shift the readout toward the rotated answer, but does not imply that the unpatched prediction run reaches the same state on its own.

5.3 Attention Allocation

Predicted rotation allocates less attention to visual tokens. We compare attention allocation across original-image direct reading, original-image predicted rotation, and rotated-image direct reading. For each condition, we group the attended context into visual, instruction, and special tokens and measure the attention mass assigned to each group across decoder layers. As shown in Figure 8, predicted rotation consistently places less attention on visual tokens and more attention on instruction tokens than the two direct-reading conditions. The two direct-reading conditions show broadly similar allocation patterns despite using different image orientations.

Summary of the diagnostic evidence. Overall, the detailed diagnostics show that prediction states can resemble rotated-image reading states even when the output favors the original string. Patching across five models further shows that states from direct rotated-image reading can shift the readout toward the rotated answer. These results do not establish that an unpatched model forms the same state or that different models follow the same internal process.

6 Conclusion

We studied *Rotated-Outcome Prediction*, which asks whether VLMs can infer the visual outcome of a 180° rotation without seeing the rotated target. With RotOutBench, we separate reading the original view, reading the rotated endpoint, and predicting that endpoint from the original view. Our results reveal a clear disconnect: many models read both visible endpoints, but not the unseen rotated one. Internal analyses further suggest that prediction states can resemble rotated-image read-

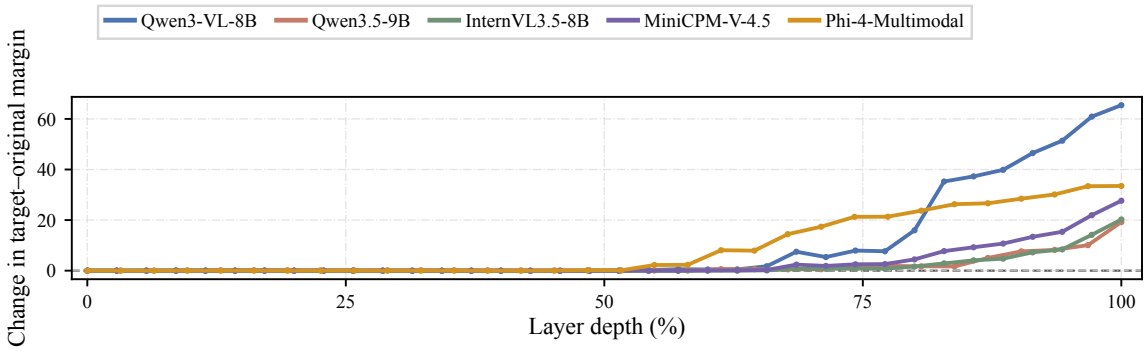

Figure 7: Patching results across five models. Curves show the change in the first-token rotated-over-original margin when the prediction run is patched with the state from direct reading of the corresponding rotated image, relative to the original-image control. Positive values indicate a stronger shift toward the rotated answer. The final plotted point is the final layer, shown only as an output-adjacent reference.

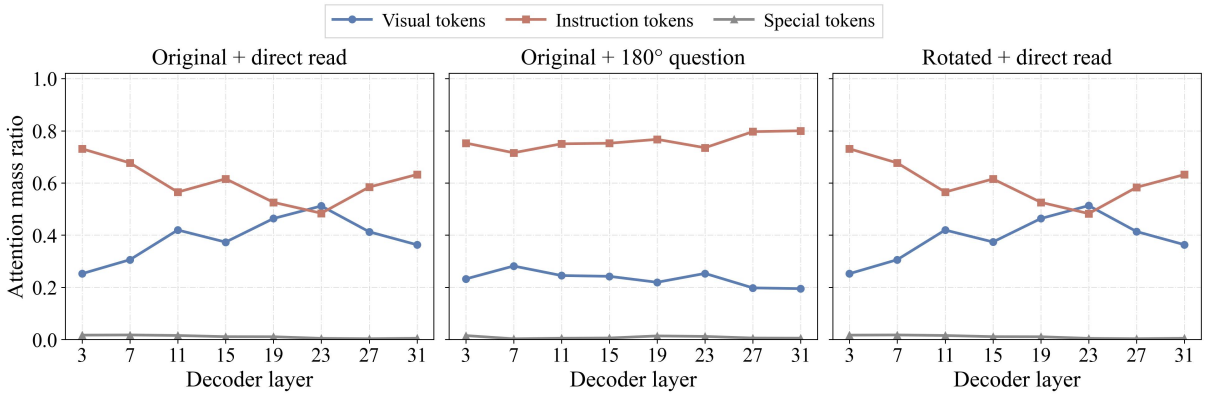

Figure 8: Attention allocation across token groups in Qwen3.5-9B. Compared with direct reading, predicted rotation allocates less attention to visual tokens and more attention to instruction tokens.

ing states even when the output favors the original string. Overall, evaluating VLMs only on visible transformed inputs can overestimate their transformation reasoning ability. Future evaluations should distinguish visible recognition from inferred visual outcomes.

7 Limitations

This work focuses on 180° in-plane rotation and a relatively small diagnostic benchmark. The current setting does not cover the full range of visual transformations or image types. Future work can extend ROTOUTBENCH to broader transformations and visual domains.

Societal impact. This work is a diagnostic study and does not introduce a deployed system. Its potential positive impact is to identify a failure mode in applications involving rotated or non-canonical visual inputs, such as document processing, assistive technologies, and embodied systems. The main risk is over-reliance on VLM inference for orientation-sensitive content without explicit verification.

Acknowledgements. This work is supported in part by the National Key R&D Program of China under Grant Nos. 2025YFC3309700, Beijing Natural Science Foundation No. 4262033, and the National Natural Science Foundation of China under Grant Nos. U25B2076, 62441229, and 62377043.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language

model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

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

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, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2025a. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, and 5 others. 2025b. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, et al. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Daniel Geng, Inbum Park, and Andrew Owens. 2024. Visual anagrams: Generating multi-view optical illusions with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24154–24163.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ji Hyeok Jung, Eun Tae Kim, Seoyeon Kim, Joo Ho Lee, Bumsoo Kim, and Buru Chang. 2025. Is 'right' right? enhancing object orientation understanding in multimodal large language models through egocentric instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14257–14267.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023. 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*, pages 9161–9175.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, and 76 others. 2025. Kimi-VL technical report. *Preprint*, arXiv:2504.07491.

Kang-il Lee, Minbeom Kim, Seunghyun Yoon, Minsung Kim, Dongryeol Lee, Hyukhun Koh, and Kyomin Jung. 2025a. Vlind-bench: Measuring language priors in large vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4129–4144.

Phillip Y Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. 2025b. Perspective-aware reasoning in vision-language models via mental imagery simulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9241–9251.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding

and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

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

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024b. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102.

Tianyi Niu, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. 2026. Rotbench: Evaluating multi-modal large language models on identifying image rotation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5569.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045.

Haz Sameen Shahgir, Khondker Salman Sayeed, Abhik Bhattacharjee, Wasi Uddin Ahmad, Yue Dong, and Rifat Shahriyar. 2024. Illusionvqa: A challenging optical illusion dataset for vision language models. *arXiv preprint arXiv:2403.15952*.

Takahiro Shirakawa and Seiichi Uchida. 2023. Ambigram generation by a diffusion model. In *International Conference on Document Analysis and Recognition*, pages 314–330. Springer.

Ilias Stogiannidis, Steven McDonagh, and Sotirios A Tsaftaris. 2025. Mind the gap: Benchmarking spatial reasoning in vision-language models. *arXiv preprint arXiv:2503.19707*.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9568–9578.

An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. 2025. Vision language models are biased. *arXiv preprint arXiv:2505.23941*.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. 2024. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Advances in Neural Information Processing Systems*, 37:75392–75421.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, and 15 others. 2025. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *Preprint*, arXiv:2509.18154.

Yichi Zhang, Jiayi Pan, Yuchen Zhou, Rui Pan, and Joyce Chai. 2023. Grounding visual illusions in language: Do vision-language models perceive illusions like humans? In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5718–5728.

Yiming Zhang, Zicheng Zhang, Xinyi Wei, Xiaohong Liu, Guangtao Zhai, and Xiongkuo Min. 2025a. Illusionbench: A large-scale and comprehensive benchmark for visual illusion understanding in vision-language models. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. 2025b. Spinbench: Perspective and rotation as a lens on spatial reasoning in vlms. *arXiv preprint arXiv:2509.25390*.

A ROTOUTBENCH Evaluation and Dataset Details

A.1 Dataset Size

Table 3 summarizes the scale of ROTOUTBENCH. A paired example contains an original view and its canonical 180° rotated view. Image views count both original and rotated views. Rotation matching is reported separately because it evaluates candidate selection rather than open-ended answering.

A.2 Input Conditions and Prompt Templates

We use X to denote the input shown to the model and Q to denote the question type. The question label alone does not determine the input. For example, Q_d only denotes a direct-reading question, and the input can be either the original image I_{orig}, the rotated target image I_{rot}, the original text string T_{orig}, or the rotated text string T_{rot}. A complete evaluation condition is therefore specified by the pair (X, Q), or equivalently by outputs such as $\hat{Y}^{Q_d}_{I_{\text{orig}}}$ and $\hat{Y}^{Q_p}_{I_{\text{orig}}}$.

For TextImage-Rot, we use fixed prompt templates. Direct reading uses Q_d:

> What string do you read? Reply with exactly the string.

This prompt is used with either the original text image I_{orig} or the rotated target image I_{rot}.

Predicted rotation uses Q_p:

> If this image were rotated 180 degrees, what string would you read? Reply with exactly the string.

This prompt is used with the original text image I_{orig}, and the reference answer is the rotated target string.

For the text-only control, we use the same question types but replace image inputs with text inputs. Direct reading uses:

> What string do you read? Reply with exactly the string.

Predicted rotation uses:

> If this string were rotated 180 degrees, what string would you read? Reply with exactly the string.

For Visual-Rot, the exact wording is image-specific because different images require different semantic targets. We therefore use Q_d and Q_p only as condition labels. A Visual-Rot direct-reading question asks about the target visible in the shown image, while a Visual-Rot predicted-rotation question asks about the corresponding target after a 180° rotation without revealing the rotated answer.

For the four-way rotation matching task, we use:

> If this image were rotated 180 degrees, which option would match? Answer with A, B, C, or D.

For the prompt-matched rotated-image control used in the internal analysis, we use Q_m:

> This image has already been rotated 180 degrees. What string do you read now? Reply with exactly the string.

This condition is paired with the rotated target image I_{rot}. It keeps rotation-related wording while making the rotated target directly visible.

For models that produce explanatory text despite the instruction, we parse the final answer using the normalization rules described below.

A.3 Answer Parsing and Normalization

For Visual-Rot, model outputs are open-ended. We lowercase the output, remove punctuation, and normalize minor formatting differences. A response is counted as correct if it matches the reference answer or one of the manually verified acceptable keywords.

The acceptable keywords are used to handle natural variation in free-form answers. They may include synonyms, visually equivalent names, common alternative names, or reasonable category-level descriptions. However, we exclude overly broad terms, visually unsupported answers, and semantically different interpretations. This prevents the evaluation from becoming overly permissive while still allowing natural variation in open-ended responses.

For TextImage-Rot, the expected answer is a specific character or string. We therefore use normalized string matching against the programmatically generated target answer. We remove surrounding whitespace and punctuation introduced by the model, but do not use semantic keyword matching for this subset.

For rotation matching, we parse the model output as an option letter. If the response contains multiple

Dataset	Paired examples	Image views	Matching questions
Visual-Rot	68	136	68
TextImage-Rot	342	684	–

Table 3: ROTOUTBENCH dataset size. A paired example contains an original view and its canonical 180° rotated view. Image views count both original and rotated views. Rotation matching is an auxiliary task with 68 four-way questions constructed from the paired Visual-Rot image views.

option letters, we use the first valid option letter unless the answer explicitly states a different final choice. Responses without a valid option letter are counted as incorrect.

A.4 Visual-Rot Review and Keyword Annotation

Each Visual-Rot example is manually reviewed to ensure that the rotated interpretation is visually grounded, that the question can be answered from the image, and that the reference answer captures the intended rotated-view interpretation. Samples are removed if the rotated view remains too ambiguous, if multiple incompatible interpretations are equally plausible, or if the answer depends on information not visible in the image.

For each retained example, we check the consistency among the original image, rotated image, question, reference answer, and acceptable keywords. The acceptable keywords are manually reviewed to include reasonable variations while excluding overly generic or visually unsupported descriptions. For example, a rotated target that can reasonably be described with a specific object name may also allow a common synonym or a visually justified category-level description, but not an unrelated object name or a generic term that does not preserve the intended visual meaning.

A.5 TextImage-Rot Generation

TextImage-Rot is generated from predefined character sets, string templates, and font styles. Each source string has a deterministic 180° rotated target. The target is produced by applying a character-level rotation mapping and reversing the character order.

For a source string

$$s = c_1 c_2 \cdots c_n,$$

its rotated target is defined as

$$\mathrm{rot180}(s) = \rho(c_n)\rho(c_{n-1}) \cdots \rho(c_1),$$

where $\rho(\cdot)$ maps each character to its 180° rotated counterpart. Characters without a well-defined rotated counterpart are excluded.

Length	Count	Generation strategy
1	6	all single characters
2	36	all 6^2 combinations
3	100	balanced sampling
4	100	balanced sampling
5	100	balanced sampling
Total	342	$6 + 36 + 100 + 100 + 100$

Table 4: String-length distribution in TextImage-Rot. Longer strings are balance-sampled to avoid combinatorial dominance while keeping the split compact and extensible.

We verify the consistency among the source string, rendered original image, rendered rotated image, and reference answer using scripts, followed by manual inspection. This ensures that the original image, rotated target image, and programmatic answer correspond to the same transformation rule.

A.6 TextImage-Rot String Distribution

TextImage-Rot uses strings of length 1 to 5. We include all available single-character and two-character combinations, and balance-sample longer strings to keep the dataset compact while avoiding dominance by any single length.

A.7 Rotation Matching Task

The four-way rotation matching task is used as an auxiliary diagnostic. It is evaluated over 68 four-way questions constructed from the paired Visual-Rot image views. Each question contains one query image and four candidate images. One candidate is the corresponding 180° rotated counterpart, while the other candidates are distractors. Candidate order is randomized.

We report the matching accuracy and its difference from the random baseline:

$$\Delta_{\mathrm{rand}} = \mathrm{Acc}_{\mathrm{match}} - 25. \tag{4}$$

Since the task has four options, the random baseline is 25%. We report this task as an auxiliary diagnostic because it evaluates candidate selection

Model	$R^{\text{text}}_{\text{orig}}$	$R^{\text{text}}_{\text{rot}}$	$R^{\text{text}}_{\text{pred}}$	G^{text}
Qwen2.5-VL-3B	92.11	92.11	1.17	90.94
Qwen3-VL-8B	**100.00**	**100.00**	0.88	99.12
Qwen3.5-9B	**100.00**	**100.00**	2.34	97.66
Gemma3-4B	**100.00**	**100.00**	1.17	98.83
Kimi-VL-A3B	**100.00**	99.71	0.88	98.83
MiniCPM-V-2.6	67.84	69.59	0.58	69.01
MiniCPM-V-4.5	**100.00**	**100.00**	**3.22**	96.78
InternVL3.5-8B	**100.00**	**100.00**	2.34	97.66
LLaVA-1.5-7B	99.12	98.54	1.17	97.37

Table 5: Text-only control results. All values are percentages. $R^{\text{text}}_{\text{pred}}$ is highlighted because predicted rotation remains close to zero even when the input is provided as plain text. The control is evaluated on models that support text-only prompts in our setup.

Model	P0	P1	P2	P3	P4
Qwen2.5-VL-3B	0.29	0.00	0.00	1.75	0.00
Qwen3-VL-8B	2.05	1.75	1.46	0.29	0.00
Qwen3.5-9B	2.34	2.92	2.34	2.63	2.34
Gemma3-4B	0.29	0.88	0.58	0.88	0.00
Kimi-VL-A3B	2.05	0.88	1.17	0.00	0.29
MiniCPM-V-2.6	0.58	0.88	0.88	1.46	0.88
MiniCPM-V-4.5	6.14	4.39	5.85	5.85	3.22
InternVL3.5-8B	2.05	2.05	1.75	1.46	1.17
LLaVA-1.5-7B	1.17	1.17	1.17	1.17	0.58

Table 6: Prompt robustness results on TextImage-Rot predicted rotation (%). P0–P4 denote the prompt variants described above.

in addition to rotated-outcome prediction. It therefore complements the open-ended prediction scores rather than serving as the main evidence.

B Additional Behavioral Results

B.1 Text-Only Control Results

The text-only control mirrors the TextImage-Rot protocol, but replaces image inputs with plain-text strings. This control asks whether the direct-reading versus prediction separation remains after removing image rendering and visual-token processing.

Let T_{orig} denote the original source string, and let T_{rot} denote the corresponding 180° rotated target string. This is the text analogue of the visual setting, where I_{orig} and I_{rot} denote the original image and the canonical rotated target image. We use Q_d for direct-reading questions and Q_p for predicted-rotation questions:

$$R^{\text{text}}_{\text{orig}} = \text{Acc}(\hat{Y}^{Q_d}_{T_{\text{orig}}}, Y_{\text{orig}}), \quad (5)$$

$$R^{\text{text}}_{\text{rot}} = \text{Acc}(\hat{Y}^{Q_d}_{T_{\text{rot}}}, Y_{\text{rot}}), \quad (6)$$

$$R^{\text{text}}_{\text{pred}} = \text{Acc}(\hat{Y}^{Q_p}_{T_{\text{orig}}}, Y_{\text{rot}}). \quad (7)$$

We also report the text-only gap:

$$G^{\text{text}} = R^{\text{text}}_{\text{rot}} - R^{\text{text}}_{\text{pred}}. \quad (8)$$

B.2 Prompt Robustness

We test whether the low predicted-rotation accuracy on TextImage-Rot is caused by a single underspecified prompt. We evaluate five prompts for the predicted-rotation condition.

P0 (Base): "If this image were rotated 180 degrees, what string would you read? Reply with exactly the string." This is the base prompt used in the main experiments.

P1 (Explicit): "The image shown is the original image, not the rotated one. If it were rotated 180 degrees, what string would be visible? Reply with only the rotated string."

P2 (Anti-copy): "Do not copy the string currently visible. Predict the string after the whole image is rotated 180 degrees. Reply with only the rotated string."

P3 (Stepwise): "First read the current string internally, then apply a 180-degree image rotation, and output only the final rotated string."

P4 (Rule-aware): "For a 180-degree image rotation, the character order is reversed and each character appears as its rotated counterpart. Apply this rule to the string in the image. Reply with only the rotated string." P4 additionally provides a high-level rotation rule and is reported as a rule-aware variant.

P1–P3 test instruction-level wording changes, while P4 gives an additional high-level rotation rule.

Table 6 shows that the failure persists across prompt variants. To further illustrate the failure patterns, Table 7 shows one representative failure example per string length. For each example, we report the model's raw output under each prompt variant, truncated to the first 20 characters. The source string and target are shown in lowercase, matching the actual image content. "–" indicates an empty output.

C Internal Diagnostic Details

C.1 Conditions and Notation

For the hidden-state, readout, and attention analyses, we focus on Qwen3.5-9B and use the TextImage-Rot setting. Let I_{orig} denote the original text image, I_{rot} denote the canonical 180° rotated target image, T_{orig} denote the original source string in text form, and T_{rot} denote the rotated target string in text form.

As in Appendix A.2, a condition is specified by both the input X and the question type Q. We use Q_d for direct reading, Q_p for predicted rotation, and Q_m for the prompt-matched rotated-image control. The wording of Q_p is instantiated as "this image" for image inputs and "this string" for text inputs.

The main image-based prediction condition is:

$$h^{Q_p}_{I_{\text{orig}}}(l),$$

where the model receives I_{orig} and is asked to predict what would be read after a 180° rotation. Here, $h(l)$ denotes the answer-position hidden state at decoder layer l.

We compare this condition with three diagnostic controls:

- $h^{Q_d}_{I_{\text{rot}}}(l)$: direct reading of the rotated target image.
- $h^{Q_m}_{I_{\text{rot}}}(l)$: prompt-matched reading of the rotated target image, where the rotated image is visible but the prompt keeps rotation-related wording.
- $h^{Q_p}_{T_{\text{orig}}}(l)$: text-only predicted rotation, where the model receives the original string as text and predicts its rotated form.

The prompt-matched control helps distinguish similarity to a visible rotated target from similarity caused only by rotation-related wording.

C.2 Hidden-State Distance

Figure 6(a) measures how close the image-based prediction state is to each control. For each decoder layer l, we compute cosine distance from $h^{Q_p}_{I_{\text{orig}}}(l)$ to the three controls:

$$D_{\text{direct}}(l) = 1 - \cos\left(h^{Q_p}_{I_{\text{orig}}}(l), h^{Q_d}_{I_{\text{rot}}}(l)\right), \quad (9)$$

$$D_{\text{matched}}(l) = 1 - \cos\left(h^{Q_p}_{I_{\text{orig}}}(l), h^{Q_m}_{I_{\text{rot}}}(l)\right), \quad (10)$$

$$D_{\text{text}}(l) = 1 - \cos\left(h^{Q_p}_{I_{\text{orig}}}(l), h^{Q_p}_{T_{\text{orig}}}(l)\right). \quad (11)$$

A smaller distance means that the image-based prediction state is more similar to that control condition. In Figure 6(a), the prediction state is closest to the prompt-matched rotated-image control $h^{Q_m}_{I_{\text{rot}}}$. This suggests that the prediction state is closer to a rotated-image reading condition than to a text-only transformation condition.

C.3 Layer-Wise Readout

Hidden-state similarity alone does not tell us which answer the model is preparing to output. A prediction state may be close to a rotated-image control, but the final answer can still be selected from a different competing representation. We therefore apply the final normalization and language modeling head to the hidden state at each layer, producing a layer-wise readout proxy.

Let Y_{orig} denote the original string answer and Y_{rot} denote the rotated target answer. For a candidate answer $Y = (y_1, \ldots, y_K)$, we compute a mean-token readout score:

$$S^{(l)}_{\text{mean}}(Y) = \frac{1}{K}\sum_{k=1}^{K} z^{(l)}(y_k), \quad (12)$$

where $z^{(l)}(y_k)$ is the logit assigned to token y_k by the layer-l readout. We also compute a first-token score:

$$S^{(l)}_{\text{first}}(Y) = z^{(l)}(y_1). \quad (13)$$

The readout margin compares the rotated target with the original string:

$$M^{(l)}_{\text{mean}} = S^{(l)}_{\text{mean}}(Y_{\text{rot}}) - S^{(l)}_{\text{mean}}(Y_{\text{orig}}), \quad (14)$$

$$M^{(l)}_{\text{first}} = S^{(l)}_{\text{first}}(Y_{\text{rot}}) - S^{(l)}_{\text{first}}(Y_{\text{orig}}). \quad (15)$$

A positive margin means that the layer-wise readout favors the rotated target. A negative margin means that it favors the original string. Figure 6(b) shows that the late-layer margin becomes negative, meaning that the readout shifts toward the original string even though the hidden state is closest to the prompt-matched rotated-image control.

C.4 Positive-Margin Fraction

To test whether the rotated target is favored for most samples, we also compute the fraction of samples with a positive readout margin:

$$F^{(l)} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[M_i^{(l)} > 0\right]. \tag{16}$$

Figure 6(c) reports this fraction across layers. A value above 0.5 means that more than half of the samples favor the rotated target at that layer. The curve does not stay above this threshold, showing that the rotated target is not selected for a stable majority of samples. This sample-level view complements the average margin in Figure 6(b): the late-layer shift is not just a mean-effect artifact, but reflects an unstable preference across examples.

C.5 Attention Grouping

For attention allocation, we group input tokens into visual tokens, instruction tokens, and special tokens. At each selected decoder layer, we measure the attention mass from the answer position to each group:

$$A_g^{(l)} = \sum_{j \in g} \alpha_{\text{ans},j}^{(l)}, \tag{17}$$

where g is a token group and $\alpha_{\text{ans},j}^{(l)}$ is the attention weight from the answer position to token j at layer l.

The text-only condition is excluded from this attention comparison because the goal is to compare how image-input conditions distribute attention between visual tokens and instruction tokens. This attention analysis is descriptive.

D Artifact Use and Implementation

D.1 Artifact Use and Intended Use

We use publicly released VLMs for research evaluation and cite their corresponding papers or technical reports. RotOutBench is intended as a diagnostic research benchmark for evaluating rotated-outcome prediction in VLMs. It should not be used as a deployed decision system or as a replacement for human verification in safety-sensitive orientation-dependent applications. For images collected from public sources, benchmark materials should be used and released according to the corresponding access conditions.

D.2 Implementation Details and Compute Budget

All experiments are inference-only. We evaluate the released model variants listed in Table 1. For each model, we use its released checkpoint and the corresponding tokenizer and processor from the official or commonly used open-source implementation. For models supported by VLMEvalKit (Duan et al., 2024), we use it for standardized inference; the remaining models are evaluated with model-specific scripts under the same prompts, decoding settings, and parsing rules. Unless otherwise stated, we use deterministic greedy decoding without sampling for short-answer evaluation.

The main behavioral evaluation, text-only control, prompt robustness experiments, and internal diagnostic analyses were run on a server with four NVIDIA A100 GPUs. The total compute budget was approximately 20 GPU-hours. The hidden-state, readout, and attention analyses are conducted on Qwen3.5-9B, while the patching analysis is evaluated across five models.

	Prompt variant			
Model	P1	P2	P3	P4
L=1: shown “q” → target “b”				
Qwen2.5-VL-3B	q	q	The letter “q” rotat…	q
Qwen3-VL-8B	q	q	q	q
Qwen3.5-9B	q	q	p	q
Gemma3-4B	Q	q	q	q
Kimi-VL-A3B	q	q	q	To solve the problem…
MiniCPM-V-2.6	The rotated string, w…	q	The final rotated strin…	The rotated string, foll…
MiniCPM-V-4.5	dq	q	q	q
InternVL3.5-8B	q	q	1	1
LLaVA-1.5-7B	A	AQ	q	Qq
L=2: shown “6b” → target “q9”				
Qwen2.5-VL-3B	b6	b6	The string “6b” is a …	6b
Qwen3-VL-8B	b6	b6	9b6	b6
Qwen3.5-9B	b9	b9	b9	b9
Gemma3-4B	b6	b6	b6	b6
Kimi-VL-A3B	6b	6b	To solve the task, we…	b6
MiniCPM-V-2.6	b6	9b	The final rotated strin…	b6
MiniCPM-V-4.5	9q	9q	9q	b9
InternVL3.5-8B	6b	6b	b6	6b
LLaVA-1.5-7B	6b	68b	68b	b68
L=3: shown “bp6” → target “9dq”				
Qwen2.5-VL-3B	6p	6p	The string “bp6” rema…	6pbb
Qwen3-VL-8B	6pob	6pob	6p6	6pob
Qwen3.5-9B	9pqb	9pqb	9pqb	9po
Gemma3-4B	6p6b	6p6b	6p b bp6	6p6b
Kimi-VL-A3B	6pmb	6pqb	To solve the problem…	6pb
MiniCPM-V-2.6	ebp6	6p9	The final rotated strin…	ebp6
MiniCPM-V-4.5	9d	9dqb	9dqb	6pbb
InternVL3.5-8B	6pb	6pb	6pb	6p6
LLaVA-1.5-7B	BPP66	BPP66	bpp66	BPP66
L=4: shown “6dq9” → target “6bp9”				
Qwen2.5-VL-3B	9q6d	9q6d6	The string “6dq9” is …	9q6
Qwen3-VL-8B	6qdl9	6q9d6	9qdl6	9qdl6
Qwen3.5-9B	6dq9	6dq9	6dq9	6dq9
Gemma3-4B	9qgd6	9qdp6	9dqp6	9qdp6
Kimi-VL-A3B	6dq9	6dq9	To solve the task, we…	9q6
MiniCPM-V-2.6	9qdd	9qdd	The final rotated strin…	9q6d
MiniCPM-V-4.5	6dq9	6dqw9	6dq9	9qdq6
InternVL3.5-8B	6dq9	6dq9	6dq9	9qd6
LLaVA-1.5-7B	DQ96	DQ96	Q96DQ96DQ96DQ96DQ96…	DQ96DQ96DQ96DQ96DQ96…
L=5: shown “9dp6b” → target “q9dp6”				
Qwen2.5-VL-3B	b6p9d9	b6p9d9	The string “9dp6b” is…	6b9dp
Qwen3-VL-8B	b6p9d	b6p9d9	b6p9d	b6p9d
Qwen3.5-9B	b9p6d	b9p6d9	b9p6d9	b9p6d9
Gemma3-4B	b6dpd9	b6pd9g	b6dp9d	b6pd9b
Kimi-VL-A3B	6bp9d	b6pd9	To solve the task, we…	b6pd9
MiniCPM-V-2.6	b6p9	b9p6d	The string “9dp6b” ro…	b6p9
MiniCPM-V-4.5	b9p6dp	b9p6d6	b9p6d6	b9p6dp6
InternVL3.5-8B	b6pd9	b6pd9	b6pd9	b6pd9
LLaVA-1.5-7B	DP6L	DP66	DP66L	DP6L

Table 7: Representative failure examples across prompt variants, organized by string length. Each row shows one model’s raw output for the same image under different prompt variants. Raw outputs are truncated to 20 characters. P0 is omitted because it uses the same prompt as the main experiment.