READING BETWEEN THE HANDS: DIAGNOSING VISUAL MEASUREMENT IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) remain unreliable at visual measurement, even as they solve harder multimodal tasks. We use analogue clock reading as a controlled setting to ask how teaching a fine-grained geometric-to-numerical conversion affects VLMs. Across models from the Qwen, Molmo, and Llama families, we combine component-specific supervised fine-tuning with controlled visual interventions, layer-wise probing, causal activation patching, and transfer to related measurement tasks. Which components benefit most from fine-tuning depends strongly on the architecture. Vision-backbone fine-tuning on synthetic clocks benefits Molmo2 most, whereas broader vision–language adaptation raises Llama from almost 0% to 85.4% accuracy and brings Qwen3.5, the strongest model in our testbed, to 92.8%. Despite these different update locations, successful adaptation often sharpens hand geometry downstream in the language pathway, even when only the vision backbone is fine-tuned. However, the size of this decodability gain does not reliably predict the accuracy gain, which suggests that more decodable geometry alone does not yield a correct readout. In Qwen3-VL, activation patching shows that hand information affects the answer mainly through visual-token states in the early and middle language layers. Finally, clock fine-tuning rarely transfers to other dial instruments, even though needle angle and scale remain linearly decodable. These results suggest that the main bottleneck is not preserving visual geometry but converting it into a reading.

1 INTRODUCTION

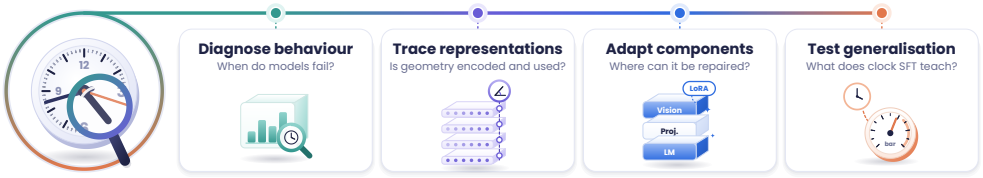


Figure 1: Overview of our diagnostic study revolving around the problem of clock reading in VLMs.

Vision-language models (VLMs) have improved rapidly on demanding multimodal tasks, including visual question answering, spatial reasoning, and multimodal reasoning (Qwen Team, 2025; Clark et al., 2026; Llama Team, 2024). They nevertheless still fail on simple visual measurement tasks. Recent work has shown that state-of-the-art multimodal models remain unreliable at reading analogue instruments and clocks (Saxena et al., 2025; Lin et al., 2025). On analogue clock reading, humans substantially outperform current VLMs: untrained human participants reach an average accuracy of 89.1% on valid-time recognition, while the best-performing closed- and open-weight models reach only 66.7% and 39.4%, respectively (Safar, 2025).

Although these benchmarks are relatively new, they already document how severe the failure is, and follow-up work has begun to address it. Intervention studies show that clock reading can be substantially improved through adaptation; early supervised fine-tuning improved a GPT-family model

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.

in-domain but did not generalise reliably beyond it (Fu et al., 2025b). Reinforcement fine-tuning produced a threefold gain, reaching 31% on synthetic instruments but 12.7% on real-world data (Lin et al., 2025), while multi-stage adaptation reached up to 46% and reduced hand confusion through Swap-DPO (Choi et al., 2026). These gains show that the skill is learnable; however, end-to-end improvement alone does not show which architectural components support it or what clock-relevant information is gained during adaptation. It remains unclear what clock-relevant information current VLMs already represent, and at which stage between that representation and the answer the reading fails.

Answering this question requires a setting in which the relevant geometry can be specified precisely and varied independently of appearance. Clocks meet both requirements, and we use them to investigate where geometric information is present in model representations, how different architectural components process, preserve, and use it, and whether this information remains usable across other dial instruments.

We study four open-weight VLMs with different multimodal architectures: Qwen3-VL-8B-Instruct, Qwen3.5-9B, Molmo2-8B, and Llama-3.2-11B-Vision-Instruct. We first measure their baseline accuracy on existing benchmarks and our DiverseSynthetic dataset, then use ControlSet, where one property varies at a time, to ask whether the same clocks are easy or hard for all models. Because no single visual factor explains the errors, we then analyse the models' internals: layer-wise probes track the hand geometry they store and propagate, CausalClock activation patching tests which information they use in the final answer, and component-specific parameter-efficient fine-tuning identifies which common VLM components support task learning and how their representations change. Finally, we test whether the learned capability generalises beyond clocks.

Our results suggest that clock-reading failures arise from how each architecture converts geometric representations into a numerical answer, and that the weak point differs across architectures.

Our contributions are as follows:

- **Controlled diagnosis of clock-reading failures.** Baseline and matched-control evaluations show architecture-specific visual sensitivities, with different models finding different clocks easy or hard.
- **Geometric and causal analysis of dial reading.** Layer-wise probes trace geometric information for clocks, pressure gauges, and ammeters, while activation patching tests how clock-hand information is causally used throughout answer generation.
- **Architecture-dependent learning routes.** Component-specific parameter-efficient fine-tuning identifies which vision, interface, and language components support task learning and how their representations change.
- **A mechanistic clock-reading testbed.** We introduce complementary datasets spanning diverse synthetic clocks, matched visual controls, hand-specific counterfactuals, equivalent textual geometry, and segmented real-world images.

2 Related Work

Clock benchmarking and learning to read clocks. Analogue clock reading predates general-purpose VLMs (Yang et al., 2022), while ClockQA, ClockBench, TickTockVQA, and MeasureBench document persistent failures in current models (Saxena et al., 2025; Safar, 2025; Choi et al., 2026; Lin et al., 2025). Adaptation establishes that the skill is learnable through synthetic supervision, supervised or reinforcement fine-tuning, and Swap-DPO, although sensitivity to dial design and real-world performance remain concerns (Deitke et al., 2025; Fu et al., 2025b; Lin et al., 2025; Choi et al., 2026). These works primarily study behaviour and mitigation rather than the representational source of failure. Controlled analyses include BlindTest's geometric judgements (Rahmanzadehgervi et al., 2024) and TriSCA's appearance interventions on clocks and gauges (Hu et al., 2026). TriSCA uses visual-feature retrieval and clustering to guide vision-tower alignment, but does not trace task-relevant geometry across layers or causally test its use. We combine matched visual controls with layer-wise hand-angle probing, causal patching, and component-specific adaptation across four architectures.

Architectural bottlenecks and adaptation targets. Prior work offers different accounts of where vision-centric bottlenecks arise. Fu et al. (2025a) found that intervening on the LLM produced the largest gains, suggesting that visual information may be available but insufficiently used. By contrast, Tong et al. (2024) linked limitations in CLIP vision encoders to downstream performance, implicating the visual backbone itself. Other studies show how LLaVA-type language models attend to, refine, or compensate for visual representations, which may nevertheless degrade in final layers (Liu et al., 2024; Neo et al., 2025; Takishita et al., 2025; Liu et al., 2025a). We extend this work by comparing component-specific fine-tuning across architectures on clock reading, a vision-centric compositional task that requires converting fine-grained geometry into a numerical answer.

Visual-information flow and causal tracing. Recent work traces visual information from image tokens through VLM decoders to linguistic and final-position representations (Neo et al., 2025; Zhang et al., 2025b), with causal tracing localising cross-modal aggregation to middle-layer computations at the last token (Li et al., 2026). Most closely, Visual Activation Patching for counting finds a handoff from distributed image-token states to the response-triggering final prompt token (Che et al., 2026). Our hand-specific clock patching supports this progressive consolidation while adding an autoregressive qualification: hand geometry remains decodable in visual-token states, but a final-token causal effect emerges only for the hour, the next generated component.

3 Experimental Setup

Models. We select four VLMs that are computationally tractable for interventions while spanning distinct multimodal architectures, vision backbones, language decoders, and fusion mechanisms. Qwen3-VL-8B-Instruct (Qwen Team, 2025), for instance, uses an MLP merger and DeepStack connections that inject intermediate visual features into the language model, whereas its successor Qwen3.5-9B (Qwen Team, 2026) retains the MLP merger but removes DeepStack. Molmo2-8B (Clark et al., 2026) uses attention pooling, while Llama-3.2-11B-Vision-Instruct (Llama Team, 2024) integrates visual information through cross-attention rather than inserting visual tokens into the initial text sequence. These and other specifications, detailed in Appendix A, provide architectural context that we return to throughout the subsequent analyses when interpreting differences across models.

Datasets. Existing clock-reading datasets (Saxena et al., 2025; Lin et al., 2025; Choi et al., 2026) do not jointly provide the visual diversity across scales, confounder-free controlled pairs, equivalent textual inputs, localisation-reduced real-world images, and causal counterfactuals required by our experiments (Table 7). We therefore introduce four complementary resources. **DiverseSynthetic**, our primary capability and fine-tuning dataset, extends the generator of Lin et al. (2025) with broader variation in clock styles, dial markings, scales, resolutions, positions, and hand arrangements. Importantly, we augment each visual configuration with two text-only formats describing the same underlying hand geometry (Appendix B.2). **ControlSet** provides confounder-free controlled subsets in which only the tested clock-face or presentation property varies, covering factors such as tick marks, numeral formats, canvas geometry, dial scale, and clock placement (Appendix B.3). **TickTockVQA-Segmented** localises and extracts clock faces from the real-world TickTockVQA dataset of Choi et al. (2026), allowing us to evaluate clock reading in real-world photographs while reducing localisation within complex scenes as a confound (Appendix B.5). Finally, **CausalClock** pairs each base clock with counterfactuals that displace the hour, minute, or second hand at multiple perturbation scales while preserving all remaining visual attributes, supporting controlled activation patching (Appendix B.4).

Evaluation. Unlike prior work that also tracks absolute error, we use the stricter *exact-match accuracy* as our primary metric: the complete time must be correct to the minute for `H:MM` examples and to the second for `H:MM:SS` examples. This reflects applications of clocks and dial instruments in high-stakes settings where precision matters; as in numerical problem solving, proximity to the correct value is not equivalent to solving the task. Further evaluation and prompting details are provided in Appendix C.1.

Table 1: Baseline exact-match accuracy (%) across clock-reading benchmarks. MB Synth. and MB Real denote the synthetic and real-world MeasureBench subsets, respectively, while TTVQA-Seg. denotes TickTockVQA-Segmented. Representative predictions are shown in Figure 12.

	DiverseSynthetic			DateTimeQA	MB Synth.	MB Real	TTVQA-Seg.
Model	H:MM	H:MM:SS	All				
GPT-5.5	51.57	41.92	45.80	30.65	47.41	32.20	14.40
GPT-5.5 (reasoning)	**66.03**	54.30	59.20	41.94	50.37	40.68	16.20
Qwen3-VL-8B-Instruct	31.10	18.21	23.60	53.23	21.11	42.37	18.40
Qwen3.5-9B	63.16	**73.20**	**69.00**	**88.71**	**80.37**	**46.61**	**38.20**
Molmo2-8B	46.89	16.84	29.40	43.55	47.78	44.92	33.00
Llama-3.2-11B-Vision-Instruct	0.00	0.00	0.00	1.61	0.00	0.85	0.20

4 Diagnosing Clock Reading

4.1 Baseline Capability

Before proceeding to a deeper diagnosis, we first verify that clock-reading failures persist across the selected range of recent VLMs. Consistent with prior work (Saxena et al., 2025; Lin et al., 2025; Choi et al., 2026), Table 1 shows that reliable analogue clock reading has not yet been achieved, although the severity of the failure varies substantially across models and datasets.

Qwen3.5 is the strongest baseline, reaching 69.0% on DiverseSynthetic and, surprisingly, outperforming proprietary GPT-5.5 even with high reasoning effort. It is particularly strong when seconds must be read (73.20% versus 63.16% on H:MM): rather than treating the second hand only as a distractor, it uses its position to refine the minute reading and avoid common $\pm$1-minute errors. In contrast, Llama remains at the floor, often collapsing to conventional guesses such as 12:00. Performance is generally lower on real-world images, especially on the varied and visually challenging TickTockVQA-Segmented examples; MeasureBench Real subset can nevertheless yield higher scores, as for Molmo2, because its examples are easier and largely follow a single wall-clock style. These results confirm that the failure is not explained solely by real-world image complexity and motivate examining which clock properties drive the model-dependent errors.

4.2 Characterising Clock-Reading Difficulty

Statistical analysis. The paired design of ControlSet holds the displayed time and all non-target attributes fixed, so each contrast isolates one factor—for example, minute ticks versus hour-only ticks. We compare paired binary exact-match outcomes using an exact McNemar test (McNemar, 1947), jointly adjust the resulting p-values using the Benjamini–Hochberg false-discovery-rate procedure (Benjamini & Hochberg, 1995), and treat contrasts with $p_{\mathrm{adj}} < 0.05$ as significant.

Summary measure. As detailed in Table 10, each attribute has multiple values, and we test every pair. For example, the three tick-mark conditions—no ticks, hour marks only, and full minute marks—yield three pairwise contrasts. Table 2 reports, for each model and attribute, the number of significant exact-match contrasts over the total number tested. A higher proportion therefore indicates greater sensitivity to changes in that attribute, but not whether those changes improve or reduce accuracy. For space, the exact accuracy differences and favoured conditions are reported in Figures 14 to 21; representative controlled pairs are shown in Figure 13.

The controlled comparisons reveal both shared constraints and pronounced model-specific structure. Surprisingly, numeral style has no significant effect for any model, providing no support for a strong OCR-based account of clock reading. Tick marks benefit the three functional models (hour ticks perform at least as well as minute ticks, and both outperform no ticks), while a visible second hand produces the largest clock-face effects, reducing exact-match accuracy by roughly 20–48 percentage points depending on the model and configuration. Displayed time is also strongly model-dependent: Qwen3-VL and Molmo2 differ significantly across 161/253 and 122/253 time pairs, respectively (253 pairwise comparisons among the 23 base times shown in Figure 7), with conventional config-

Table 2: Summary of controlled sensitivities. Cells report the number of significant pairwise exact-match contrasts over all contrasts for each attribute; shading is proportional to this fraction.

	Clock attributes				Presentation factors		
Model	Numerals	Tick marks	Second hand	Displayed time	Canvas size	Position	Dial scale
Qwen3-VL	0/10	2/3	6/10	161/253	2/21	15/36	2/3
Qwen3.5	0/10	2/3	6/10	36/253	7/21	8/36	2/3
Molmo2	0/10	2/3	4/10	122/253	1/21	7/36	0/3
Llama	0/10	0/3	1/10	22/253	0/21	0/36	0/3

urations such as 3:00, 10:10, and 9:15 often easier than near-hour transitions such as 2:59 and 8:57; Qwen3.5 is more stable at 36/253. Position preferences likewise differ: Qwen3-VL and Qwen3.5 tend to perform worse toward the top of the image, whereas Molmo2 exhibits a more localised weakness at the bottom-right. Thus, the models do not share a universal notion of an easy clock, but rely on different visual regularities when reading one.

4.3 Visual Reading Does Not Reduce to Explicit Geometric Conversion

To separate visual extraction from numerical conversion, we evaluate the same DiverseSynthetic clock configurations as images or as *text-only* degree-based and clock-native descriptions (Table 9). Explicit hand positions do not guarantee an accurate readout: all models perform worse on text-only inputs than on images (Table 3), with the near-complete collapse on degree-based questions reflecting the need to convert continuous angles before composing the time. Clock-native descriptions reduce this burden and recover the hour especially well, but still yield weak minute and second accuracy (Table 15). Test-time reasoning partially reverses this pattern for Qwen3.5, raising exact match from 3% to 62% on degree descriptions and from 17% to 73% on clock-native descriptions, yet sharply reducing image accuracy from 57% to 13% (Table 16), consistent with prior reports that reasoning can harm vision-centric tasks (Kancheti et al., 2026; Sammani et al., 2026). Its image traces reveal a striking repetitive self-correction loop: the model repeatedly re-identifies the hands, revises previously plausible readings, and can exhaust even an 8,192-token budget without settling on a final answer. This failure is not universal, as GPT-5.5 improves from 49% to 63% on images with high reasoning effort; the full setup, results, and traces are reported in Appendix C.3.

Table 3: Exact-match accuracy (%) by input condition. Hand-level results are reported in Appendix C.2.

Model	Image	Text (deg.)	Text (native)
Qwen3-VL	23.6	0.2	25.8
Qwen3.5	69.0	1.6	22.0
Molmo2	29.4	0.6	17.8
Llama	0.0	0.0	2.2

5 Representation Learning: Where Does the Geometry Go?

To determine where clock geometry becomes linearly accessible and whether this accessibility persists throughout the network, we train layer-wise linear probes to predict the angles of the hour, minute, and second hands. We target continuous latent geometry—the angles—rather than discrete time readings, thereby separating geometric representation from answer-token preparation in the language decoder. Details of the circular target representation and probe fitting are provided in Appendix E. Linear decodability alone does not establish that the model causally uses this information, which we examine next. Conversely, failed decoding would not establish that the information is absent: the model may encode or access it through mechanisms that a linear probe cannot recover (Ravichander et al., 2021; Fu et al., 2025a).

In the base Qwen3-VL-8B-Instruct (Figure 2), hand geometry becomes progressively more accessible through the visual pathway and remains decodable after multimodal fusion, indicating no clear projector bottleneck. However, the lowest errors still correspond to roughly 6–9 minute marks: minute- and second-hand angles are less precise than the hour-hand angle, and second-hand decodability declines in the final language layers. Thus, poor clock reading cannot be explained by a wholesale loss of hand geometry.

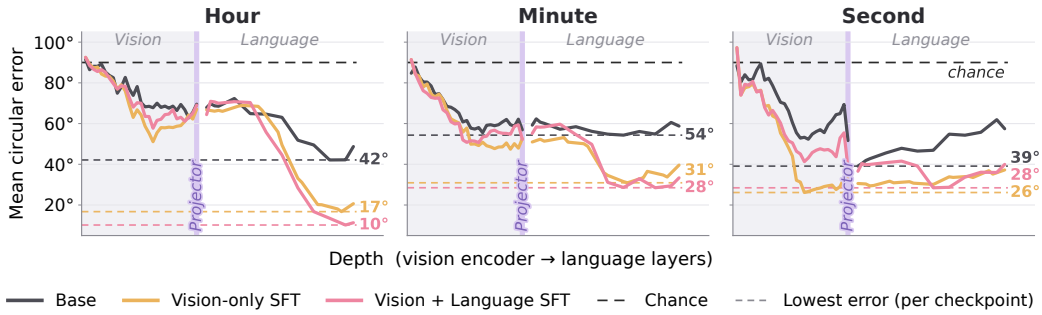


Figure 2: Layer-wise hand-angle probe error for Qwen3-VL-8B-Instruct before and after component-wise fine-tuning. Lower is better; the black dashed line marks the 90° random baseline, and coloured dashed lines mark checkpoint minima. Improvements occur mainly in the language pathway, even after vision-only tuning, with additional second-hand gains in the vision encoder.

This broad pattern recurs across architectures, with informative differences. In Molmo2 (Figure 23), decodability initially drops at its attention-pooling projector, which may weaken local spatial context (Cha et al., 2024). Unlike the other models, however, its second-hand decodability improves through the language decoder and remains strong near the output. Its abstractor-style projector or local image tiles may contribute, but the probes cannot distinguish these mechanisms. In Llama (Figure 24), the vision encoder follows the Qwen3-VL trend but with substantially higher error, while language-layer decodability remains near chance, consistent with its near-floor behavioural accuracy.

5.1 From Decodability to Causal Use

To test whether the model uses, rather than merely encodes, hand geometry, we perform causal activation patching on CausalClock image pairs that differ in one hand by a controlled displacement. During a run on the base image, we replace activations at a selected layer and token scope with those from the counterfactual run and measure the resulting shift toward the counterfactual answer. Our metric, the *fraction of the counterfactual effect recovered*, uses answer log probabilities and therefore captures shifts even when the generated answer does not change (Appendix F).

Figure 3 shows similar causal trajectories across the three hands. Patching vision-token states redirects the model toward the counterfactual reading through the early and middle language layers, but the effect largely disappears in the final layers. Patching instruction-token states has negligible effect, indicating that the counterfactual clock state is not broadly copied into the textual stream.

In the final layers, the visual-token effect gives way to an effect at the final encoded input token, but only for hour-hand perturbations, consistent with the autoregressive handoff described by Che et al. (2026). This token's state directly predicts the first answer component, and the model generates `HH`→`MM`→`SS`; the minute and second therefore do not immediately follow the patched prompt token. Supporting this interpretation, patching the corresponding last-token state immediately before minute and second generation yields recovery comparable to that observed for the hour. The trajectory is also stable across displacement scales: magnitude-dependent differences in the vision encoder largely diminish in the language decoder (Figure 26). Thus, hand geometry is causally conveyed mainly through vision-token states until the middle language layers, after which the last-token states used for autoregressive prediction become central.

6 Locating Effective Adaptation Targets

The probes show that geometric information can remain partially accessible even when final reading accuracy is low. We therefore ask which architectural components can be adapted to improve clock reading and, subsequently, how these interventions alter geometric decodability. We fine-tune on DiverseSynthetic, which was designed to provide broad variation in clock appearance and configuration. We apply LoRA (Hu et al., 2021) to the vision encoder, multimodal interface, language

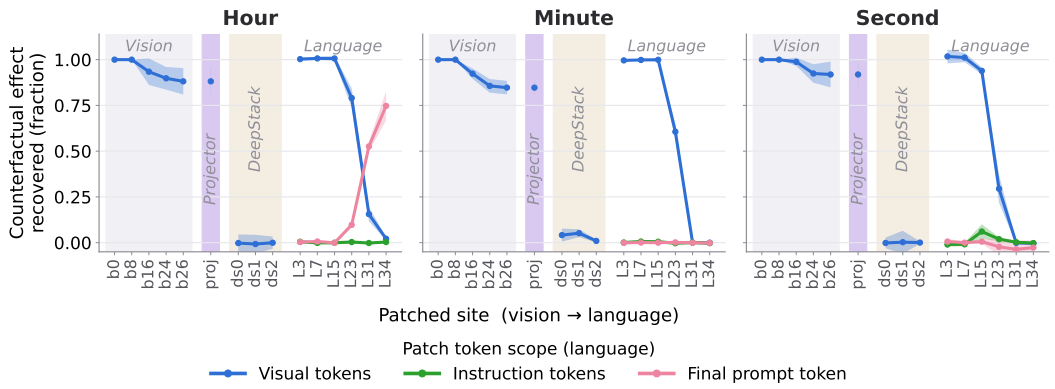


Figure 3: Counterfactual-effect recovery from activation patching across Qwen3-VL-8B-Instruct depth for single-hand perturbations. Values of 0 and 1 denote no recovery and full recovery, respectively; shading shows 95% confidence intervals. Vision-token patches recover nearly the full effect in early-to-middle language layers, whereas instruction-token patches have little effect. Only hour-hand perturbations affect the final prompt token, consistent with the hour being generated first.

decoder, and selected combinations while freezing all other parameters. We use a common configuration rather than matching trainable parameter counts, so the interventions compare adaptation routes rather than per-parameter efficiency; full details are provided in Appendix G. The best route is strongly architecture-dependent (Figure 5). Vision–language adaptation raises Qwen3-VL from 23.6% to 84.8% and Llama from 0.0% to 85.4%, whereas vision-only adaptation is best for Molmo2, improving accuracy from 29.4% to 71.0%. Qwen3.5 starts at 69.0% and reaches 89–93% across all interventions. Thus, there is no universal component-level bottleneck, and the most effective update target need not coincide with where geometry is least accessible. In Qwen3-VL, hand-angle decodability is weakest in the early vision backbone and improves toward multimodal fusion (Figure 2), yet language-inclusive adaptation yields the largest behavioural gain.

6.1 How Fine-Tuning Changes Representations and Sensitivities

However, it is not clear whether task learning sharpens the geometric information needed to ground the answer and which model components play the most important role. The best-performing vision–language Qwen3-VL-8B-Instruct checkpoint substantially improves hand-angle decodability in the language pathway. To determine whether this effect requires language-model updates, we also probe the vision-only checkpoint and observe similar downstream sharpening (Figure 2). The second hand is the main exception: vision-only tuning also produces a clear improvement within the vision encoder, consistent with its finer spatial demands.

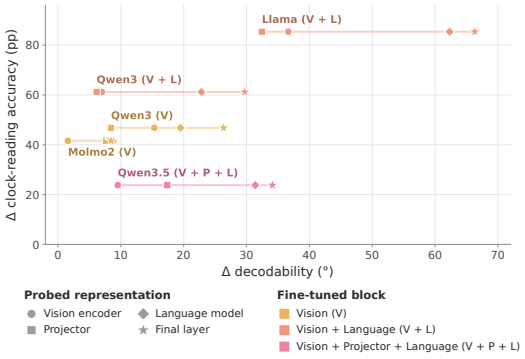


Figure 4: Change in clock-reading accuracy versus hand-angle decodability after fine-tuning, measured relative to each base model.

This comparison separates three locations that need not coincide: where geometry is already decodable, where fine-tuning changes its decodability, and where parameter updates most improve end-to-end behaviour. In the aggregate comparison in Figure 4, each block is represented by its lowest mean circular error across probed layers, averaged over the three hands, with changes measured relative to the base model. The magnitude of improvement in downstream decodability does not always correlate with the accuracy gain, a pattern also visible in the hand-wise breakdown in Figure 25. This suggests either that models draw

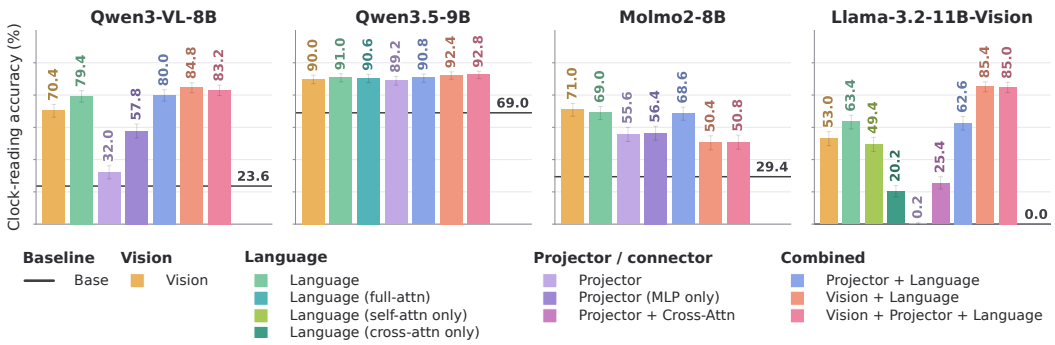


Figure 5: Exact-match accuracy on DiverseSynthetic after component-specific LoRA fine-tuning; horizontal lines mark pre-fine-tuning performance. The best adaptation target is model-dependent and typically combines multiple components. See Table 19 for complete results.

on other accessible information beyond geometric placement or that additional mechanisms govern how geometric information is applied in the procedural conversion to time readouts.

Further post-SFT checks in Appendix D.3 show that fine-tuning also reorganises the models' sensitivity profiles. For three of the four models, sensitivity to the displayed time falls sharply, and previously favourable canonical times such as 10:10 and 3:00 largely lose their advantage. This is consistent with the baseline effect partly reflecting training-data bias towards canonical clock configurations. The change is not uniformly stabilising, however, as some presentation sensitivities (e.g., canvas size) become stronger.

Together, these analyses show that fine-tuning does not repair a single localised deficit. Instead, it improves behaviour through architecture-specific update routes, reshapes where geometry is accessible, and redistributes rather than uniformly removes visual sensitivities. The mismatches among these effects argue against treating any one component as the sole clock-reading bottleneck.

7 Generalisation Beyond Clocks

7.1 Non-Clock Dial Instruments

Beyond in-domain gains, we ask whether clock-reading fine-tuning transfers to related measurement tasks. Clocks are unusually complex pointer-based instruments, typically requiring three hands to be distinguished and converted into a reading; learning this structure might therefore make simpler single-pointer gauges easier to interpret. We evaluate every clock-fine-tuned checkpoint on eight non-clock dial instruments from MeasureBench and compare its instrument and clock-reading accuracy. Contrary to this hypothesis, clock fine-tuning does not transfer broadly (Table 4 and Figure 27). Across models, instruments, and adaptation targets, fine-tuned checkpoints match or underperform their untuned baselines in the large majority of cases despite substantial clock-reading gains. Improvements are especially rare when the vision backbone is tuned: vision-inclusive configurations exceed baseline performance in approximately 6% of instrument evaluations, compared with approximately 25% for configurations tuning only the connector and/or language model.

Table 4: Exact-match accuracy (%) of Qwen3-VL-8B before and after component-specific LoRA fine-tuning on clocks; parentheses show changes from the base model. Instrument rows pool test domains. See Appendix H for all models and components.

			Component fine-tuned on clocks		
Task	n	**Base**	**Vision**	**Language**	**Vision + Lang.**
Clock	500	23.6	70.4 (+46.8)	79.4 (+55.8)	84.8 (+61.2)
Ammeter	156	16.0	5.8 (-10.2)	9.6 (-6.4)	9.0 (-7.0)
Compass	57	5.3	5.3 (0.0)	10.5 (+5.2)	8.8 (+3.5)
Hygrometer	52	15.4	1.9 (-13.5)	13.5 (-1.9)	13.5 (-1.9)
Pressure Gauge	238	12.2	4.2 (-8.0)	8.8 (-3.4)	6.3 (-5.9)
Fuel Gauge	16	25.0	12.5 (-12.5)	12.5 (-12.5)	6.3 (-18.7)
Speedometer	42	21.4	7.1 (-14.3)	31.0 (+9.6)	21.4 (0.0)
Voltmeter	30	13.3	3.3 (-10.0)	10.0 (-3.3)	6.7 (-6.6)
Weighing Scale	60	18.3	8.3 (-10.0)	15.0 (-3.3)	6.7 (-11.6)
Synthetic	330	13.0	4.5 (-8.5)	10.3 (-2.7)	6.1 (-6.9)
Real-World	321	15.6	5.9 (-9.7)	13.1 (-2.5)	11.5 (-4.1)
All instruments	651	14.3	**5.2** (-9.1)	**11.7** (-2.6)	**8.8** (-5.5)

Further probing suggests that weak behavioural transfer is not caused by missing instrument information. Pressure-gauge and ammeter needle angles are decodable from vision-encoder activations with approximately 2–3° error, while the ammeter's unit and range remain almost perfectly decodable after clock fine-tuning (Figure 28 in Appendix H). Although this establishes accessibility rather than causal use or a shared readout, both pointer position and scale identity are available. A dial reading can be written as $r = r_{\min}(c) + g(\theta, c)[r_{\max}(c) - r_{\min}(c)]$, where c identifies the scale, $r_{\min}$ and $r_{\max}$ are its endpoints, and g maps the needle angle to a relative scale position. For clocks, g is fixed—for example, $m = \theta_m/6$ for the minute hand—whereas gauge zero points, ranges, angular spans, directions, and tick spacing vary. The likely bottleneck is therefore composing accessible angle and scale information through this image-dependent mapping: clock fine-tuning learns a nearly fixed conversion, not the family of scale-conditioned conversions required by other instruments.

7.2 Broader spatial reasoning tasks

Beyond transfer to other instruments, we ask whether clock fine-tuning affects broader spatial-reasoning capabilities. Table 5 shows that the vision–language checkpoint with the highest clock-reading accuracy improves slightly on SpatialMQA (Liu et al., 2025b), but yields a modest aggregate decline on SPHERE (Zhang et al., 2025a). This aggregate result masks variation across skills: performance improves on some tasks, including object manipulation with intermediate states, but degrades on tasks involving counting. Qualitative SPHERE examples reveal no consistent visual pattern distinguishing corrected predictions from newly introduced errors (Figures 29 and 30), while the aggregate results provide no evidence of broad degradation across this diverse set of spatial skills.

Table 5: Spatial generalisation of Qwen3-VL after vision–language LoRA fine-tuning on DiverseSynthetic. Values are accuracy (%); parentheses show changes from the base model, and SPHERE scores are micro-averaged.

Benchmark	**Base**	**Vision + Lang.**
SpatialMQA	36.2	38.2 (+2.0)
SPHERE (overall)	69.3	67.8 (-1.5)
Single-Skill	79.2	77.6 (-1.6)
Multi-Skill	62.6	61.0 (-1.6)
Reasoning	64.3	63.0 (-1.3)

8 Conclusion

Starting from a simple failure case in VLMs, we extend prior work on diagnosis and fine-tuning with a *multifaceted diagnosis* of the visual measurement procedure centred on clock reading, rather than seeking a single explanation for its failure. The baseline evaluation first confirms that analogue clock reading remains unresolved across a diverse model testbed, while the controlled study shows that there is no universal notion of an "easy" clock: models favour different visual configurations, and even seemingly useful cues such as numerals do not systematically improve reading. At the representation level, hand geometry remains modestly but consistently decodable, while causal patching shows that its influence is carried predominantly through visual-token states and evolves with the autoregressive structure of the output, with the earliest generated component becoming output-facing before the others. Component-wise fine-tuning then shows that clock reading can be improved through multiple architecture-dependent routes, with substantial gains often accompanied by sharper geometric representations in the language pathway even when the updated parameters lie elsewhere. Yet these gains rarely transfer to other dial instruments, despite pointer angle and scale information remaining readily accessible, suggesting that clock fine-tuning primarily sharpens a clock-specific procedure for converting geometry into a numerical readout rather than learning a general-purpose measurement skill. The limited changes on broader spatial tasks further suggest that this specialisation is relatively targeted rather than a wholesale degradation of visual-spatial capability. A natural next step is therefore to characterise the language-side mechanisms that implement this geometric-to-readout conversion and to test whether a more flexible scale-conditioned procedure can be learned without sacrificing transfer across measurement systems.

AI use statement

Generative AI tools were used to assist with several aspects of the research and manuscript preparation. They were used for proofreading and improving the clarity of the writing; generating and formatting LaTeX tables from author-provided experimental results; assisting with the preparation

and refinement of scientific figures; and assisting with software code development. AI-assisted code was reviewed, tested, and verified by the authors before use in the reported experiments. Generative AI tools were also used to assist with searching for and identifying relevant literature for parts of the related-work review; cited sources were subsequently inspected and verified by the authors. In addition, generative AI was used as a research assistant for discussion and interpretation of some experimental results and for feedback on aspects of experimental design. The authors independently evaluated the resulting analyses, experimental decisions, and claims. We reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, results, code, figures, and citations.

Reproducibility statement

We provide detailed descriptions of the evaluated models and architectural components in Appendix A, dataset construction and preprocessing in Appendix B, prompting and inference settings in Appendix C.1, probing methodology in Appendix E, causal activation patching in Appendix F, and fine-tuning configurations in Appendix G. Dataset sizes, train/test protocols, hyperparameters, and additional per-model results are reported throughout the corresponding appendices. An anonymized repository containing the experimental code, evaluation scripts, prompts, and configurations required to reproduce the reported results is provided with the submission.

References

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 57(1):289–300, 1995. ISSN 0035-9246. URL http://www.jstor.org/stable/2346101.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13817–13827, June 2024.

Liwei Che, Zhiyu Xue, Yihao Quan, Benlin Liu, Zeru Shi, Michelle Hurst, Jacob Feldman, Ruixiang Tang, Ranjay Krishna, and Vladimir Pavlovic. Counting circuits: Mechanistic interpretability of visual reasoning in large vision-language models, 2026. URL https://arxiv.org/abs/2603.18523.

Jaeha Choi, Jin Won Lee, Siwoo You, and Jangho Lee. It's time to get it right: Improving analog clock reading and clock-hand spatial reasoning in vision-language models, 2026. URL https://arxiv.org/abs/2603.08011. Version Number: 1.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, April 2026. URL http://arxiv.org/abs/2601.10611. arXiv:2601.10611 [cs].

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat,

Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross B. Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 91–104. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00018. URL https://openaccess.thecvf.com/content/CVPR2025/html/Deitke_Molmo_and_PixMo_Open_Weights_and_Open_Data_for_State-of-the-Art_CVPR_2025_paper.html.

Stephanie Fu, Tyler Bonnen, Devin Guillory, and Trevor Darrell. Hidden in plain sight: VLMs overlook their visual representations. In *Second Conference on Language Modeling*, 2025a. URL https://openreview.net/forum?id=qQb1JLrwol.

Tairan Fu, Miguel González, Javier Conde, Elena Merino-Gómez, and Pedro Reviriego. Have multimodal large language models really learned to tell the time on analog clocks? *IEEE Internet Computing*, 29(4):48–54, 2025b. doi: 10.1109/MIC.2025.3618144.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yuanze Hu, Gen Li, Yuqin Lan, Qingchen Yu, Zhichao Yang, Junwei Jing, Zhaoxin Fan, and Xiaotie Deng. State beyond appearance: Diagnosing and improving state consistency in dial-based measurement reading. *CoRR*, abs/2604.26614, 2026. doi: 10.48550/ARXIV.2604.26614. URL https://doi.org/10.48550/arXiv.2604.26614.

Sai Srinivas Kancheti, Aditya Sanjiv Kanade, Vineeth N. Balasubramanian, and Tanuja Ganu. Chain-of-thought degrades visual spatial reasoning capabilities of multimodal LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 862–876, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-391-3. doi: 10.18653/v1/2026.acl-short.71. URL https://aclanthology.org/2026.acl-short.71/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Qiming Li, Zekai Ye, Xiaocheng Feng, Weihong Zhong, Weitao Ma, and Xiachong Feng. Causal tracing of object representations in large vision language models: Mechanistic interpretability and hallucination mitigation. In *AAAI Conference on Artificial Intelligence*, pp. 31645–31653, 2026. doi: 10.1609/AAAI.V40I37.40431. URL https://arxiv.org/abs/2511.05923.

Fenfen Lin, Yesheng Liu, Haiyu Xu, Chen Yue, Zheqi He, Mingxuan Zhao, Miguel Hu Chen, Jiakang Liu, JG Yao, and Xi Yang. Do vision-language models measure up? benchmarking visual measurement reading with MeasureBench. *arXiv preprint arXiv:2510.26865*, October 2025. URL http://arxiv.org/abs/2510.26865. arXiv: 2510.26865.

Benlin Liu, Amita Kamath, Madeleine Grunde-McLaughlin, Winson Han, and Ranjay Krishna. Visual representations inside the language model. *ArXiv*, abs/2510.04819, 2025a. URL https://arxiv.org/abs/2510.04819.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296, June 2024. doi: 10.1109/CVPR52733.2024.02484.

Jingping Liu, Ziyan Liu, Zhedong Cen, Yan Zhou, Yinan Zou, Weiyan Zhang, Haiyun Jiang, and Tong Ruan. Can multimodal large language models understand spatial relations? In

Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 620–632, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.31. URL https://aclanthology.org/2025.acl-long.31/.

Llama Team. The Llama 3 herd of models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs].

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chanJGoa7f.

Qwen Team. Qwen3-VL technical report, November 2025. URL http://arxiv.org/abs/2511.21631. arXiv:2511.21631 [cs].

Qwen Team. Qwen3.5: Towards native multimodal agents. Qwen Blog, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind. In Minsu Cho, Ivan Laptev, Du Tran, Angela Yao, and Hongbin Zha (eds.), *Computer Vision - ACCV 2024 - 17th Asian Conference on Computer Vision, Hanoi, Vietnam, December 8-12, 2024, Proceedings, Part V*, volume 15476 of *Lecture Notes in Computer Science*, pp. 293–309. Springer, 2024. doi: 10.1007/978-981-96-0917-8_17. URL https://doi.org/10.1007/978-981-96-0917-8_17.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard H. Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 3363–3377. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.295. URL https://doi.org/10.18653/v1/2021.eacl-main.295.

Alek Safar. CLOCKBENCH: Visual time benchmark where humans beat the clock, LLMs don't. https://clockbench.ai/ClockBench.pdf, September 2025. Published as Alek Safar (Oleg Chichigin), accessed 2026-04-15.

Fawaz Sammani, Tzoulio Chamiti, and Nikos Deligiannis. On test-time scaling for vision-language models. In *European Conference on Computer Vision*, pp. 168–185, 2026. doi: 10.1007/978-3-032-37271-0_10. URL https://arxiv.org/abs/2606.28864.

Rohit Saxena, Aryo Pradipta Gema, and Pasquale Minervini. Lost in time: Clock and calendar understanding challenges in multimodal LLMs. *arXiv preprint arXiv:2502.05092*, March 2025. URL http://arxiv.org/abs/2502.05092. arXiv: 2502.05092.

Sho Takishita, Jay P. Gala, Abdelrahman Mohamed, Kentaro Inui, and Yova Kementchedjhieva. LLMs can compensate for deficiencies in visual representations. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15253–15272, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.825. URL https://aclanthology.org/2025.findings-emnlp.825/.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578, June 2024. doi: 10.1109/CVPR52733.2024.00914.

Charig Yang, Weidi Xie, and Andrew Zisserman. It's about time: Analog clock reading in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2498–2507, April 2022. doi: 10.1109/CVPR52688.2022.00254. URL http://arxiv.org/abs/2111.09162. arXiv:2111.09162 [cs].

Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Junqi Zhao, Allison Koenecke, Boyang Li, and Lu Wang. SPHERE: Unveiling spatial blind spots in vision-language models through hierarchical evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11591–11609, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.568. URL https://aclanthology.org/2025.acl-long.568/.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19781–19791, 2025b. doi: 10.1109/CVPR52734.2025.01842.

A Model Suite: Architectures and Parameter Counts

A.1 Architectural Details

We focus on the architectural features that determine how visual information enters the language pathway and how the component-specific interventions in Section 6 are defined.

Qwen3-VL-8B-Instruct. The model uses a 27-layer vision transformer and a 36-layer Qwen3-8B language decoder. Its main visual merger compresses each 2×2 group of vision features into one token and projects it into the 4,096-dimensional language-model space using layer normalisation and a two-layer GELU MLP. DeepStack additionally extracts features from vision-encoder layers 8, 16, and 24. Dedicated mergers project these representations and inject them residually into the first three decoder layers, giving early language layers access to intermediate as well as final visual features.

Qwen3.5-9B. Qwen3.5 retains a similar 27-layer vision encoder and 2×2 MLP merger but removes the DeepStack pathway. Its principal architectural difference is a 32-layer hybrid language decoder that repeats three Gated DeltaNet layers followed by one full self-attention layer eight times. The sequential stack therefore contains 24 linear-attention layers and eight full-attention layers rather than separate parallel branches.

Molmo2-8B. Molmo2 combines a 25-layer SigLIP 2 vision encoder with a 36-layer Qwen3-8B language decoder. A global crop is processed alongside overlapping higher-resolution crops, using up to $K = 8$ crops during training and $K = 24$ at inference. The connector combines features from the third-to-last and ninth-from-last vision layers, reduces each 2×2 patch window through 16-head attention pooling, and maps the result into the 4,096-dimensional language space with an MLP projector. Column tokens preserve the spatial arrangement of the multi-crop representation.

Llama-3.2-11B-Vision-Instruct. Llama 3.2 Vision extends a pretrained Llama 3.1 language model with a separately trained vision encoder and cross-attention adapter. Images are resized and arranged into fixed-size tiles according to their aspect ratio, then processed through local and global transformer stages. Rather than inserting visual tokens directly into the text sequence, 8 cross-attention layers are interleaved through the decoder. At these layers, language hidden states query the encoded visual representations, allowing image information to be retrieved repeatedly at multiple decoder depths.

A.2 Model and Trainable Parameter Counts

Table 6 reports each model's total parameter count, the size of its principal architectural components, and the number of parameters made trainable under rank-16 LoRA adaptation. Component size is distinct from the number of updated parameters: the former describes the complete targeted module, whereas the latter counts only the LoRA adapters attached to eligible layers. Combined configurations are omitted because their counts can be obtained by summing the corresponding component rows.

The interventions are controlled by a shared LoRA configuration rather than an exactly matched trainable-parameter budget. In contrast, Fu et al. (2025a) restrict the eligible weight matrices to equalise the number of trainable parameters across components. Our protocol instead applies the same targeting rules throughout each selected component, allowing the adapter count to scale with component size. The rankings in Section 6 therefore identify effective adaptation routes under a common training recipe; they do not constitute parameter-matched causal attribution, intrinsic capacity comparisons, or estimates of per-parameter efficiency.

Table 6: Full architectural component sizes and trainable parameter counts for the evaluated models. Full component counts include all parameters in the corresponding architectural block. Trainable counts report the rank-16 LoRA adapters updated during fine-tuning; LoRA is applied only to eligible modules within the named component. Shares are calculated relative to the complete model.

		Full component		Trainable LoRA, $r = 16$	
Model	Component	Parameters	Model share	Parameters	Model share
Qwen3-VL-8B	*Complete model*	**8.77B**	100.00%	–	–
	Language model	7.57B	86.33%	43.65M	0.498%
	Vision encoder	415.89M	4.74%	7.70M	0.088%
	Projector + DeepStack	160.50M	1.83%	1.15M	0.013%
	Projector MLP	40.12M	0.46%	286.72K	0.003%
	LM head	622.33M	7.10%	–	–
Qwen3.5-9B	*Complete model*	**9.41B**	100.00%	–	–
	Language model	7.94B	84.34%	43.28M	0.460%
	Full-attention modules	469.77M	4.99%	3.93M	0.042%
	Vision encoder	415.89M	4.42%	7.70M	0.082%
	Projector	40.12M	0.43%	286.72K	0.003%
	LM head	1.02B	10.81%	–	–
Molmo2-8B	*Complete model*	**8.66B**	100.00%	–	–
	Language model	7.57B	87.38%	36.57M	0.422%
	Vision encoder	383.03M	4.42%	8.05M	0.093%
	Projector	87.94M	1.02%	894.98K	0.010%
	Projector MLP	78.64M	0.91%	692.22K	0.008%
	LM head	622.33M	7.18%	–	–
Llama-3.2-11B-V	*Complete model*	**10.67B**	100.00%	–	–
	Language model	9.25B	86.69%	52.43M	0.491%
	Decoder layers (excl. cross-attn.)	8.39B	78.62%	49.02M	0.459%
	Decoder layers (cross-attn.)	335.55M	3.14%	3.41M	0.032%
	Vision encoder	863.57M	8.09%	14.75M	0.138%
	Projector	31.46M	0.29%	188.42K	0.002%
	LM head	525.34M	4.92%	–	–

B Dataset Construction and Validation

B.1 Existing Clock-Reading Datasets

Table 7 summarises the existing clock-reading resources used in this study. They differ not only in image source and size, but also in their annotation formats: TickTockVQA uses single time strings, DateTimeQA provides exact times including seconds, and MeasureBench represents acceptable readings as intervals.

Dataset	Synthetic	Real-World	Example labels
TickTockVQA (Choi et al., 2026)	–	7,236 / 5,247	`01:03 PM`; `10:08 AM`; `06:20 None`
DateTimeQA (Saxena et al., 2025)	62	–	`4:00:00`; `12:35:05`; `1:00:00`

MeasureBench (Lin et al., 2025)	270	118	`["5:11", "5:13"]`
			`["12:52", "12:54"]`
			`[["08:59:06", "08:59:07"],`
			`["20:59:06", "20:59:07"]]`

Table 7: Overview of the existing clock-reading datasets considered in this study. TickTockVQA counts are reported as train/test; the other entries give the total number of examples of each image type. The final column illustrates the different answer representations rather than listing every possible label format.

B.2 DiverseSynthetic Construction

DiverseSynthetic broadens the visual distribution of the original MeasureBench synthetic-clock generator through four rendering regimes (Table 8). The regimes vary dial geometry, colour, text, clock placement, cropping, and background while preserving exact access to the underlying hand angles and displayed time.

Table 8: The four style regimes used by the DiverseSynthetic generator.

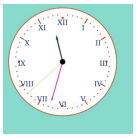
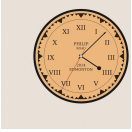
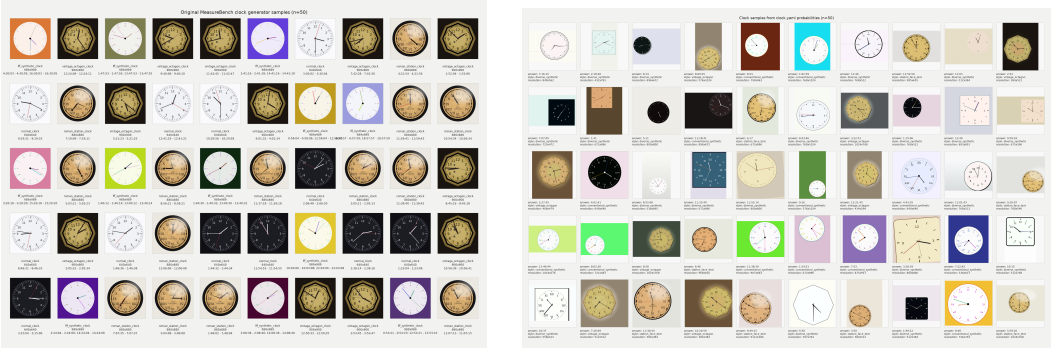
Style	Description	Example
`diverse_synthetic`	High-diversity clocks with broad variation in colour palettes, hand colours, shapes and thicknesses, ticks, numerals, dial appearance, resolution, placement, and mild rotation. The placement distribution includes edge, corner, off-centre, and mildly cropped clocks.	
`conventional_synthetic`	Cleaner benchmark-like clocks with conventional faces, clearer contrast, and conservative layouts. Most clocks are centred or only mildly displaced.	
`vintage_octagon`	Decorative wall clocks with vintage-inspired colours, octagonal or framed dials, textured backgrounds, and muted wall- or wood-like settings. This regime introduces non-circular outer geometry while retaining a readable dial.	
`station_face_text`	Station-style clocks containing non-answer text such as names, dates, station labels, or location-like inscriptions on the face.	

Table 9 illustrates the three question forms generated for the same underlying clock image: direct image reading, degree-based hand conversion, and clock-native hand-position conversion.

Figure 6 contrasts samples from the original MeasureBench generator with the diversified generation pipeline. The new pipeline increases variation in face design and presentation rather than merely adding more samples from the original rendering distribution.



(a) Original MeasureBench pipeline **(b) Diversified pipeline**

Figure 6: Fifty clock images sampled from the original MeasureBench generation pipeline and fifty images sampled from the diversified pipeline used to construct DiverseSynthetic.

B.3 ControlSet Construction

ControlSet is built from 23 base times organised into seven intended difficulty and hand-geometry groups (Figure 7). Each comparison holds the displayed time and all non-tested attributes fixed, allowing changes in accuracy to be associated with a single controlled factor.

The Clock Face subset varies properties of the dial itself, whereas the Presentation subset fixes the face design and varies how the clock is placed within the image (Table 10). Compared with prior controlled clock resources, this design covers both hand configuration and a broader range of clock-face and presentation attributes.

Figure 8 illustrates the five second-hand regimes used in both subsets. A small random angular offset prevents the regimes from becoming fixed templates.

The full Cartesian product for the Clock Face subset would contain $23 \times 5 \times 3 \times 5 = 1{,}725$ images. Some second-hand regimes are not physically distinct for every base time: for example, when the hour and minute hands overlap, positions near either hand coincide. Generating only meaningful configurations yields 1,335 images. The Presentation subset begins from $23 \times 5 \times 7 \times 3 \times 9 = 21{,}735$ requested combinations. Near-duplicate off-centre placements are removed when the dial is too large for the requested displacement to produce a meaningful positional change, leaving 13,915 images. Each example records the displayed time, second-hand regime, canvas size, aspect class, dial scale, requested position, realised clock centre, and displacement from the centred position.

B.4 CausalClock Construction

CausalClock is designed for hand-specific activation patching. Source images are sampled from the four DiverseSynthetic families used by our synthetic renderer: diverse synthetic (probability 0.40), conventional synthetic (0.20), vintage octagonal (0.20), and station-style clocks (0.20). Resolution, theme, dial geometry, placement, background, bezel, ticks, numerals, and hand designs are sampled once for each source clock and then held fixed across all of its counterfactuals. Hours and minutes are sampled uniformly, and a second hand is included with probability 0.5; seconds are sampled uniformly when it is present.

Each counterfactual intervenes on exactly one hand while preserving the face, rendering configuration, and all non-target hand angles. The intervention therefore represents an isolated geometric change rather than elapsed clock time. Perturbations are grouped by their displacement around the corresponding 12- or 60-unit dial (Table 11). Local interventions move by one unit in a randomly

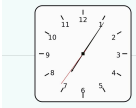
Image	Question	Text-only question	Trivial text-only question	Answer
	Read the analog clock. If a second hand is visible, return the time as H:MM:SS. If no second hand is visible, return the nearest minute as H:MM. Return only the time.	The hour hand is rotated 212.8 degrees clockwise from 12 o'clock. The minute hand is rotated 33.6 degrees clockwise from 12 o'clock. The second hand is rotated 216.0 degrees clockwise from 12 o'clock. What time is it? Return only the time in 12-hour H:MM:SS format.	The hour hand is between 7 and 8, closer to 7, the minute hand points at 1, and the second hand points 1 tick after 7. What time is it? Return only the time in 12-hour H:MM:SS format.	`7:05:36`
	Read the analog clock. If a second hand is visible, return the time as H:MM:SS. If no second hand is visible, return the nearest minute as H:MM. Return only the time.	The hour hand is rotated 205.2 degrees clockwise from 12 o'clock. The minute hand is rotated 302.6 degrees clockwise from 12 o'clock. There is no second hand visible on the clock. What time is it? Return only the time in 12-hour H:MM format.	The hour hand is between 6 and 7, closer to 7, and the minute hand points at 10. There is no second hand visible on the clock. What time is it? Return only the time in 12-hour H:MM format.	`6:50`

Table 9: Examples of the image, degree-based text-only, and clock-native text-only question forms generated for DiverseSynthetic. Each row presents three equivalent descriptions of the same clock and their shared answer.

Table 10: Attributes varied in the Clock Face and Presentation subsets of ControlSet.

Attribute	Values
Clock Face Control Set	
Base times	23 times from seven difficulty and hand-geometry groups; see Figure 7.
Numerals	All Arabic; all Roman; Arabic quarter numerals only; Roman quarter numerals only; no numerals.
Tick marks	No ticks; hour marks only; full minute marks.
Second-hand regime	Absent; near the minute hand; near the hour hand; near both hands; far from both hands.
Resolution and position	Fixed at 512×512, with the clock centred.
Visual design	Minimalist black-and-white face with a red second hand.
Presentation Control Set	
Base configuration	The same 23 times and five second-hand regimes; Arabic numerals and full minute ticks are held fixed.
Canvas size	384×384; 512×512; 768×768; 768×512; 512×768; 1024×512; 512×1024.
Dial scale	0.45; 0.65; 0.85 of the shorter canvas side.
Clock position	Centre; left; right; top; bottom; and the four corners.

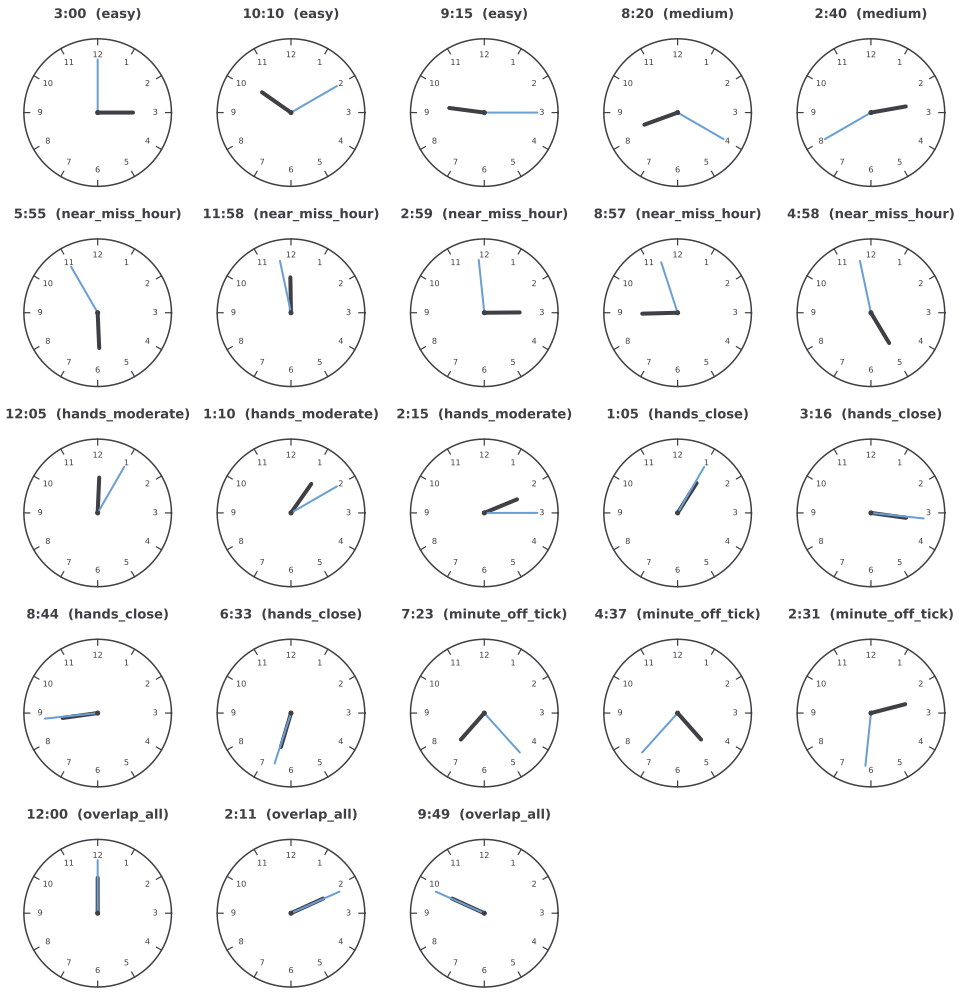


Figure 7: The 23 base times used to construct ControlSet, grouped by intended difficulty and hand geometry.

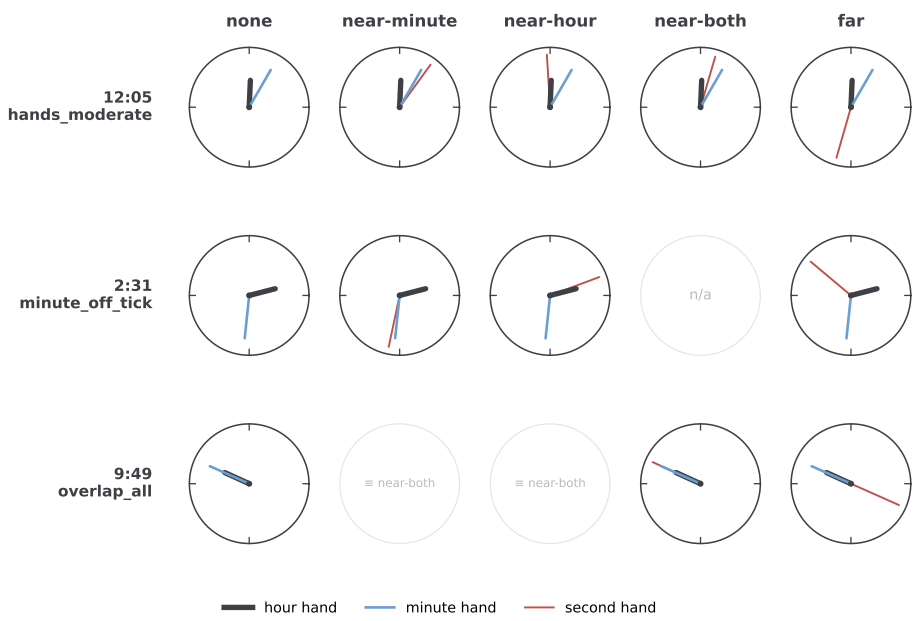


Figure 8: The five second-hand regimes used in ControlSet: absent, near the minute hand, near the hour hand, near both hands, and far from both hands.

sampled direction; Q1–Q4 interventions move clockwise. Values wrap around the dial independently, with no carry propagated to another hand. For example, moving a minute hand from 55 by $+10$ produces minute 5 without changing the hour hand.

Table 11: Hand-specific perturbation bands used to construct CausalClock counterfactuals. Local direction is sampled uniformly between clockwise and anticlockwise; Q1–Q4 are applied clockwise.

Perturbation band	Hour-hand displacement	Minute/second displacement
Local	exactly 1 hour	exactly 1 unit
Q1	2–3 hours	2–15 units
Q2	4–6 hours	16–30 units
Q3	7–9 hours	31–45 units
Q4	10–11 hours	46–59 units

For source clocks with a visible second hand, five perturbations are generated for each of the three hands, yielding one base image and 15 counterfactuals. For clocks without a second hand, hour- and minute-hand perturbations are generated as above, and Q1–Q4 additionally introduce a visible second hand; these sources yield one base image and 14 counterfactuals. The release uses random seed 42 and contains 100 source clocks, split evenly by initial second-hand visibility, for a total of 1,550 images (100 bases and 1,450 counterfactuals). Each record stores source and counterfactual times and angles, the target hand, perturbation band and signed displacement, second-hand visibility, rendering attributes, image geometry, a source-pair identifier, and the complete rendering configuration.

B.5 TickTockVQA-Segmented Construction and Validation

We derive TickTockVQA-Segmented from TickTockVQA using the SAM3 (Carion et al., 2025) checkpoint distributed by Meta. General clock images are prompted with `analog clock`, whereas wristwatch images use `watch`, which improves detection of wristwatch samples. Candidate detections below 0.5 object confidence are discarded, masks are binarised at 0.5, and the highest-confidence remaining detection is retained. The foreground bounding rectangle is padded

by 25% of its width and height on each side to preserve the outer tick ring and bezel. If the padded crop covers more than 70% of the original image, the uncropped image is retained.

Table 12 reports the initial extraction outcomes. Across the original train and test pools, SAM3 successfully processes 97.8% of images.

Table 12: SAM3 extraction outcomes before construction of the final TickTockVQA-Segmented splits. Retained uncropped denotes successful images preserved pixel-identically under the 70% rule.

Split	Initial	Successful	No detection	Retained uncropped
Train	7,236	7,054 (97.48%)	182 (2.52%)	2,366 (33.54%)
Test pool	5,247	5,155 (98.25%)	92 (1.75%)	739 (14.34%)

We manually audit 200 images sampled across the training, development, and test partitions, stratified by split, source dataset, and clock type. Of the 142 images for which a crop is produced, the complete clock face is retained in every case and the intended target is confidently selected in 141 cases (99.3%; Table 13). The remaining 58 images are retained uncropped because the clock already occupies more than 70% of the image.

Table 13: Manual quality audit of SAM3 preprocessing on 200 sampled TickTockVQA images. Percentages for indented rows are calculated relative to their parent outcome.

Audit outcome	**Count**	**Percentage**
All audited samples	**200**	**100.0%**
Segmented crop produced	**142**	**71.0%**
Correct target selected	141	99.3%
Target assessment uncertain	1	0.7%
Complete clock face retained	142	100.0%
Uncropped image retained	**58**	**29.0%**
Clock occupied more than 70% of image	58	100.0%

The audit separately assesses whether the original source image is readable. Among the 142 cropped instances, 129 (90.8%) are confidently suitable for determining the displayed time (Table 14). Unsuitable examples primarily contain clocks that are too small, insufficiently resolved, or occluded; these limitations originate in the source images rather than in the SAM3 crop.

Table 14: Manual audit of original-image quality for the 142 sampled TickTockVQA instances that received a segmented crop.

Original-image quality outcome	**Count**	**Percentage**
Hour hand clearly visible	133	93.7%
Hour hand at least identifiable	138	97.2%
Minute hand clearly visible	135	95.1%
Minute hand at least identifiable	138	97.2%
Image confidently suitable for reading	129	90.8%

Cropping changes the resolution distribution of the source data (Figure 9). Figure 10 shows paired source and segmented images, while Figure 11 shows examples for which neither prompt produced a detection.

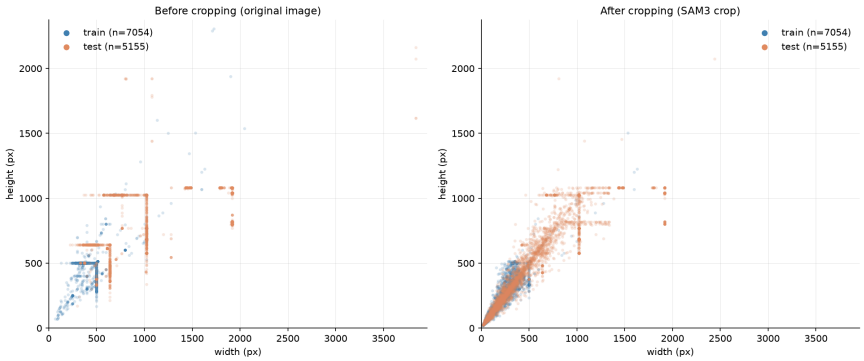


Figure 9: Image-resolution distributions for the original TickTockVQA images and the corresponding TickTockVQA-Segmented outputs after SAM3 segmentation, padding, and application of the uncropped-image rule.

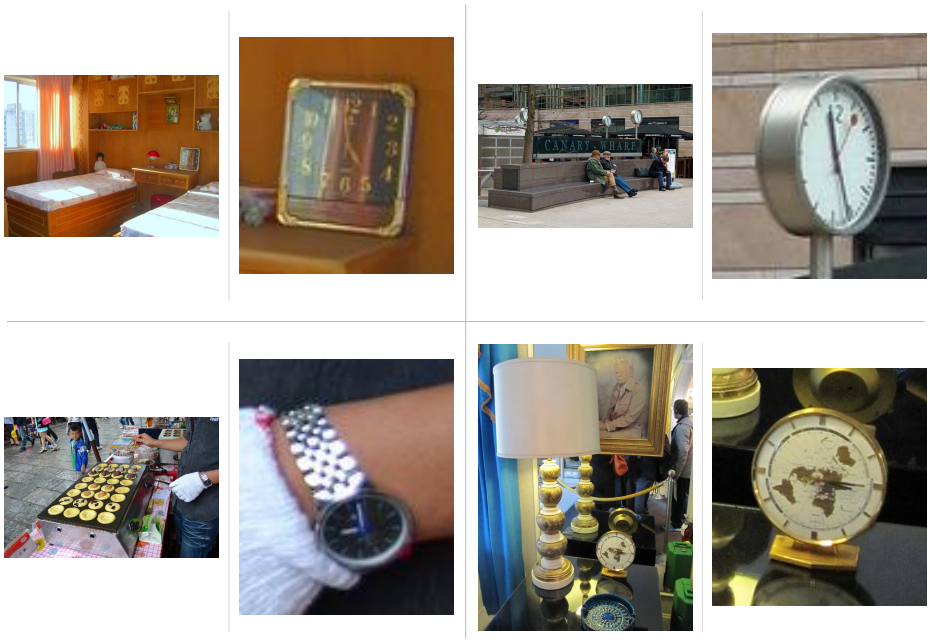


Figure 10: Examples of original TickTockVQA images and the corresponding segmented outputs produced by SAM3.

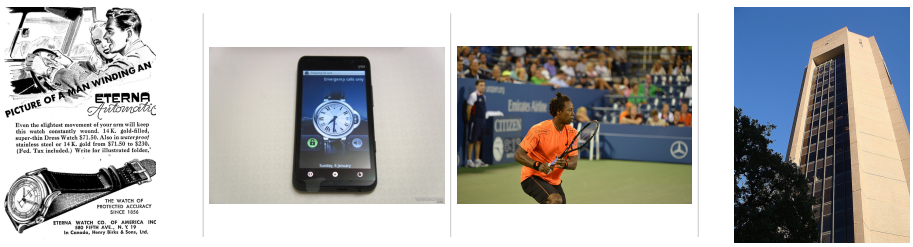


Figure 11: Examples of TickTockVQA images for which SAM3 produced no detection using either the `analog clock` or `watch` prompt.

C Baseline Predictions and Reasoning Experiments

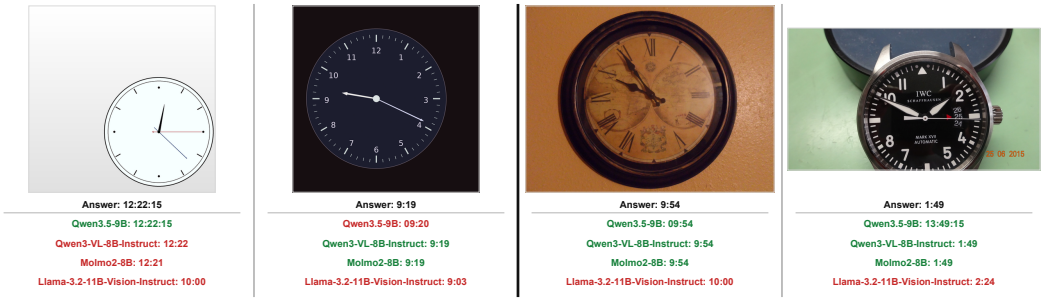
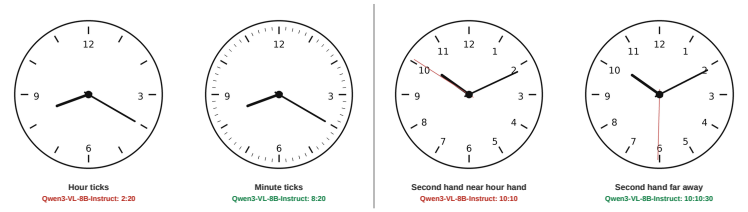
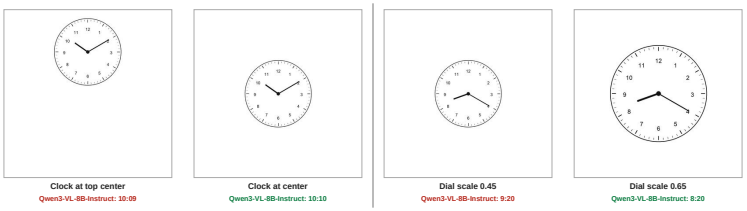


Figure 12: Representative examples and predictions from the four open-weight models evaluated in Table 1. The first two samples are from DiverseSynthetic and the final two are from TickTockVQA-Segmented.



(a) Clock-face factors



(b) Presentation factors

Figure 13: Matched examples of Qwen3-VL-8B-Instruct sensitivities corresponding to the controlled comparisons summarised in Table 2. Paired images differ only in the tested attribute.

C.1 Prompt and Inference Protocol

Unless stated otherwise, model evaluations use the vLLM inference engine (Kwon et al., 2023) in bfloat16 precision with deterministic decoding: `temperature=0.0`, `top_p=1.0`, and `top_k=-1`. Examples are submitted individually with `batch_size=1`, simplifying per-sample output alignment and error analysis. Standard evaluations request a direct answer without an explicit reasoning trace.

Main clock-reading prompt

```
Read the analog clock.  If a second hand is visible, return
the time as H:MM:SS. If no second hand is visible, return the
nearest minute as H:MM. Return only the time.
```

The conditional format requires the model to determine whether a second hand is present rather than revealing the expected answer format in advance. This preserves the visual distinction between two-hand and three-hand clocks and prevents the prompt from supplying information that should be inferred from the image.

Molmo direct-answer evaluation. For Qwen3.5, explicit reasoning is disabled with `enable_thinking=False`. Molmo2 has no equivalent reasoning toggle and is not a dedicated reasoning model, but it sometimes produces step-by-step arithmetic on degree-based text questions despite the instruction to return only the time. To obtain direct, consistently scorable responses in this condition, we prefill the assistant turn with `The time is` and set `add_generation_prompt=False`, so generation continues from an answer prefix rather than beginning with an explanation. This prefilling is used only where spontaneous arithmetic otherwise prevents comparable direct-answer evaluation.

C.2 Text-Based Questions

Table 15 reports the hand-level results corresponding to the exact-match comparison in Table 3. The three conditions present the same underlying clock configurations as an image, as ground-truth hand angles in degrees, or as clock-native textual descriptions.

Table 15: Exact-match and hand-level accuracy (%) on DiverseSynthetic by model and input condition. Second accuracy is evaluated on examples containing a visible second hand.

Model	Input condition	Exact	Hour	Minute	Second
Qwen3-VL	Image	23.6	73.0	55.4	36.8
	Text (degrees)	0.2	12.4	4.0	4.5
	Text (clock-native)	25.8	100.0	46.6	25.4
Qwen3.5	Image	69.0	97.6	75.8	83.5
	Text (degrees)	1.6	42.8	7.0	13.8
	Text (clock-native)	22.0	100.0	38.0	18.2
Molmo2	Image	29.4	96.4	57.8	25.4
	Text (degrees)	0.6	20.4	5.2	7.9
	Text (clock-native)	17.8	99.6	32.6	27.1
Llama	Image	0.0	13.0	2.0	1.7
	Text (degrees)	0.0	15.2	10.8	6.2
	Text (clock-native)	2.2	86.2	7.4	2.4

C.3 Test-Time Reasoning Experiments

We test whether additional inference-time computation can recover the explicit geometric conversion ability measured under the image, degree-based text, and clock-native text conditions. For Qwen3.5, direct answering uses `enable_thinking=False`, while reasoning uses `enable_thinking=True` with a 2,048-token generation budget; we additionally test 8,192 tokens for the trace and omission analysis. For GPT-5.5, we compare no reasoning with high reasoning effort. Table 16 summarises the exact-match results on a fixed 100-example subset of the DiverseSynthetic test set.

Table 16: Effect of test-time reasoning on exact-match accuracy (%) on a fixed 100-example subset of the DiverseSynthetic test set. Component-level results and generation-budget comparisons are reported in Table 17.

	GPT-5.5			Qwen3.5		
Input/Reasoning	Off	On	Δ	Off	On	Δ
Image	49	63	+14	57	13	-44
Text (deg.)	97	97	0	3	62	+59
Text (native)	100	100	0	17	73	+56

Additional reasoning improves symbolic conversion but degrades image-based reading in Qwen3.5: enabling reasoning raises exact-match accuracy from 3% to 62% for degree-based descriptions and from 17% to 73% for clock-native descriptions, but lowers image-reading accuracy from 57% to

13%. This degradation is consistent with prior work suggesting that inference-time reasoning procedures developed for text-only LLMs do not transfer uniformly or monotonically to vision-centric tasks (Kancheti et al., 2026; Sammani et al., 2026). Qualitative traces reveal repeated re-evaluation of hand identities and positions; a larger generation budget sometimes reduces second-hand omissions but does not resolve the overall image-reading deficit. The effect is not universal: GPT-5.5 improves from 49% to 63% on images with high reasoning effort. Together, these results are consistent with models relying on a learned image-to-answer route rather than a mandatory perceive–then–compute pipeline.

Table 17: Effect of test-time reasoning on DiverseSynthetic. Results are rounded to the nearest percentage point and reported on a fixed 100-example subset containing 46 `H:MM` and 54 `H:MM:SS` examples. For Qwen3.5, max tokens denotes the generation budget when reasoning is enabled. The skipped-second-hand rate is the percentage of `H:MM:SS` examples for which the second hand is omitted and is reported only for image inputs.

Model	Input	Reasoning	Max tokens	Exact match			Hour			Minute			Second			Skipped second hand (%)
				H:MM	H:MM:SS	All	H:MM	H:MM:SS	All	H:MM	H:MM:SS	All	H:MM	H:MM:SS	All	H:MM:SS
GPT-5.5	Image	None	–	52	46	49	83	93	88	54	54	54	80	74	77	7
	Image	High	–	67	59	63	96	94	95	67	69	68	93	83	88	6
	Text (degrees)	None	–	93	100	97	100	100	100	93	100	97	100	100	100	–
	Text (degrees)	High	–	93	100	97	100	100	100	93	100	97	100	100	100	–
	Text (clock-native)	None	–	100	100	100	100	100	100	100	100	100	100	100	100	–
	Text (clock-native)	High	–	100	100	100	100	100	100	100	100	100	100	100	100	–
Qwen3.5	Image	Off	–	50	63	57	98	100	99	52	70	62	96	83	89	6
	Image	On	2,048	17	9	13	65	83	75	22	28	25	96	15	52	65
	Image	On	8,192	22	7	14	72	80	76	24	26	25	98	20	56	30
	Text (degrees)	Off	–	2	4	3	39	48	44	9	15	12	100	19	56	–
	Text (degrees)	On	2,048	41	80	62	100	94	97	52	91	73	67	81	75	–
	Text (clock-native)	Off	–	30	6	17	100	100	100	30	35	33	100	15	54	–
	Text (clock-native)	On	2,048	85	63	73	85	87	86	85	87	86	89	63	75	–

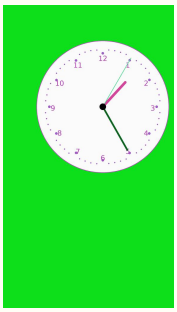
C.3.1 Qualitative Analysis of Reasoning Traces

Sampling configuration. For the Qwen3.5 thinking experiments, we use `temperature`=1.0, `top_p`=0.95, `top_k`=20, `presence_penalty`=1.5, and `repetition_penalty`=1.0, following the Qwen3.5-9B model card's recommendations for thinking mode on general tasks.[1] Preliminary experiments with deterministic decoding (`temperature`=0) yielded the same exact-match accuracy as the corresponding sampling runs. We nevertheless use the recommended sampling configuration for the reported thinking experiments.

Effect of the generation budget. We compare selected excerpts from Qwen3.5 image-reading runs with generation budgets of 2,048 and 8,192 tokens on the same examples. In both examples, the shorter run exhausts its budget while still assessing the second hand, leaving a last time prediction that omits seconds. With the larger budget, repeated re-evaluation eventually yields a time prediction that includes the second-hand value. The skipped-second-hand column in Table 17 quantifies this pattern: the omission rate rises from 6% with thinking disabled to 65% at 2,048 tokens, then falls to 30% at 8,192 tokens while remaining well above the direct-answer baseline. These examples therefore illustrate reduced second-hand omission at the larger budget, rather than a general improvement in exact-match accuracy. The last time predictions are reproduced from the example summaries. Excerpts retain the original wording, including inconsistencies, and are separated by ellipses where intervening text is omitted.

[1] `https://huggingface.co/Qwen/Qwen3.5-9B`

Example 1: Resolving the second-hand position



Ground truth: `1:25:05`

Last time predictions:
2,048 tokens: `1:25`
8,192 tokens: `1:25:05`

2,048-token excerpts
Red highlights mark repeated re-evaluation; blue marks the last predicted time; bracketed ellipses mark omitted passages.

The user wants me to read the time from the provided image of an analog clock.
*1. **Analyze the clock face:***
• The hour markers are purple numbers 1 through 12.
• There are tick marks for minutes.
• There is a center hub.
*2. **Identify the hands:***
*• **Hour hand:** There is a thick pink/magenta hand pointing towards the 1 o'clock position, but slightly past it. Wait, let's look closer. It's pointing towards the 1.*
*• **Minute hand:** There is a long green hand pointing towards the 25-minute mark (between 4 and 5, specifically on the 5th minute mark which corresponds to the 25). Wait, let's re-evaluate the positions.*
[. . .]
Let's re-examine carefully.
*• **Hand 1 (Short, thick magenta):** Pointing directly at the 1. This would suggest 1:00.*
*• **Hand 2 (Long, dark green):** Pointing towards the 25-minute mark (number 5). This suggests 25 minutes past.*
*• **Hand 3 (Thin, light green/cyan):** Pointing towards the 7-minute mark (between 1 and 2). Wait, let's look closer. It's pointing near the 1.*
[. . .]
Let's try a different interpretation.
• Maybe the magenta hand is the minute hand? If it's at 1, that's 5 minutes.
• Maybe the green hand is the hour hand? If it's at 25, that's near 5. That doesn't make sense. Hour hand moves slowly.
[. . .]
• So, Hour = 1, Minute = 25.
• Is there a second hand? Yes, the thin cyan hand. Where is it pointing?
• It is pointing towards the 7 minute mark? No, let's look at the tip. It's pointing near the 1. Wait, looking really closely at crop 1 and the main image.
[. . .]
Let's test the hypothesis: Time is 5:05.
[. . .]
Let's look at the lengths again to distinguish hour vs minute.
Usually, the hour hand is shorter.
The magenta hand seems shorter than the dark green hand.
The dark green hand reaches all the way to the number 5 (or slightly past).
The magenta hand reaches towards the number 1.
So, Magenta = Hour, Green = Minute.
Time: 1:25.
Now, what about the cyan hand?
It is a thin line. It looks like a second hand.
Where is it pointing?

[Generation ends here.]

8,192-token excerpts

The user wants me to read the time on the analog clock.
*1. **Analyze the clock face:***
• It's a standard 12-hour clock face.
• The numbers are 1 through 12, arranged clockwise.
• There are small dots between the numbers representing the minute markers.
• The hour hand is the shorter, thicker hand. It's pinkish-red.
• The minute hand is the longer, thinner hand. It's dark green.
• There is a very thin, light green hand pointing near the 12/1 area. Wait, let me look closer.
[. . .]
• Let's re-evaluate the hand pointing to the 1. It's pink.
• Let's re-evaluate the hand pointing to the 5. It's dark green.
• Let's re-evaluate the thin light green hand. It points towards the 12/1 boundary. It looks like a second hand. If the time is roughly 1:25, the second hand could be anywhere.
[. . .]
• If the time is 1:25, and the second hand is at 1, that means 5 seconds.
• So the time would be 1:25:05.
• Let's check if the second hand is visible. Yes, there is a thin light green line.
• Let's check the position. It points to the number 1.
• So seconds = 5.
• So the time is 1:25:05.
• Wait, let me look really closely at the hands again.
[. . .]
• Let's reconsider the hand identification.
• Maybe the pink hand is the minute hand? It looks somewhat thick.
• Maybe the dark green hand is the hour hand? It looks long and thin.
• If dark green is hour hand pointing to 5.
• If pink is minute hand pointing to 1 (5 mins).
• Then time is roughly 5:05.
[. . .]
• Final check.
• Hour hand: Pink, short. Points to 1.
• Minute hand: Dark Green, long. Points to 5.
• Second hand: Light Green, thin. Points to 1.
• Time: 1:25:05.
• Wait, is the light green hand pointing to the 12?
• If it points to 12, seconds = 00.
• If it points to 1, seconds = 05.
[. . .]
• Let's try to verify the second hand position more precisely.
• The number 1 is at roughly 15 degrees (clockwise from 12).
• The cyan line seems to point to roughly 15 degrees.
• So 5 seconds.
• Final conclusion:
• Hour: 1
• Minute: 25
• Second: 5
• Format: 1:25:05.
• Wait, let's look at the second hand again. Is it possible it's pointing to the 11?
• No, it's clearly in the top right quadrant.
• Is it possible it's pointing to the 55

[Generation ends here.]

Interpretation. The 2,048-token run repeatedly revisits hand identities and positions, considers alternative times, and returns to `1:25` before questioning the cyan hand again. It discusses possible second-hand values but ends without incorporating seconds into its last extractable time; the extraction metadata identifies an hour–minute time line as the answer source. The 8,192-token run incorporates seconds into `1:25:05`, but repeatedly reopens hand identification and second-hand estimation, even after a "Final check" and "Final conclusion". Its last stated time includes seconds despite the continued re-evaluation and inconsistent angle estimate. This example illustrates reduced second-hand omission, not the elimination of repetitive reasoning.

Example 2: Separating overlapping hands



Ground truth: `12:36:36`

Extracted time predictions:
2,048 tokens: `12:35`
8,192 tokens: `12:36:36`

2,048-token excerpts

Red highlights mark repeated re-evaluation; blue marks the extracted time; bracketed ellipses mark omitted passages.

The user wants me to read the time from the provided analog clock image.

*1. **Analyze the hour hand (short, thick black hand):***
- *It is pointing past the 12.*
- *It is approaching the 1.*
- *It looks like it's around the 12:40 or 12:45 position. Wait, let's look closer.*
- *The hour hand is pointing towards the 12 but has moved past it. Wait, no. Let's re-evaluate.*

[. . .]

*2. **Analyze the minute hand (longer black hand):***
- *It is pointing towards the 7.*
- *Wait, let's look at the numbers. 12 is at the top. 6 is at the bottom.*
- *The longer black hand is pointing towards the 7. It looks like it's pointing exactly at the 7. So the minutes would be 35 (since 7 * 5 = 35).*
- *Let's re-examine the hands.*

[. . .]

- *Actually, let's look at the red hand. That's a second hand.*
- *Let's re-evaluate the black hands.*
- *There is a hand pointing towards the 12. It's thick.*
- *There is a hand pointing towards the 7. It's long.*
- *Let's reconsider the positions.*

[. . .]

- *So the time is 12:35.*
- *Now let's look at the red hand (second hand).*
- *It is pointing near the 7. Wait, looking closely at the bottom left.*
- *There is a black hand pointing towards the 7. Wait, no.*
- *Let's look really closely at the hands.*

[. . .]

- *Hand C: Red thin hand. Points towards the 7 as well, but slightly further counter-clockwise. It's pointing near the tick mark for roughly 28 seconds? Or is it pointing near the 7 (35 seconds)?*
- *Let's re-evaluate the positions.*

• *Maybe the thick hand is the minute hand? No, that's unlikely.*
• *Maybe the hand pointing to the 7 is the hour hand? If the hour hand is at 7, the time is 7:something. Then the minute hand (the other black one) must be near 12.*
• *Let's look at the "other black hand". It is pointing towards the 12.*
• *If the time is 7:00, the hour hand is at 7 and minute hand is at 12.*

[Generation ends here.]

8,192-token excerpts

The user wants me to read the time from the provided analog clock image.

1. ***Analyze the hour hand:*** *The short, thick black hand is pointing past the 12, but not yet at the 1. It's slightly past the 12, maybe around 12:03 or so. Wait, let me look closer. It's pointing towards the 12, but definitely past it. It's pointing roughly at the position between 12 and 1, closer to 12. Actually, looking at the angle, it's pointing towards the 12, specifically past the 12 mark. Let's re-evaluate.*

[. . .]

• *The red hand is the second hand. It is pointing towards the 6 and 7 area. It's pointing at roughly the 36th second mark (just past the 7).*

[. . .]

• *Let's restart the hand identification.*
• *Hand A: Short, thick black. Points towards 12, slightly past. This is the Hour hand. Time is 12:something.*
• *Hand B: Long, black. Points towards 7. Wait, let's look at the tip. It's pointing towards the 7. Wait, is it pointing at the 36 minute mark?*

[. . .]

• *So, Minute hand is at ˜36 minutes. Second hand is at ˜36 seconds?*
• *That would mean the time is 12:36:36.*
• *Let's check the position of the minute hand again. Is it at 36 minutes?*
• *The number 7 is at 35 minutes. The hand is pointing just past it. Yes, looks like 36 minutes.*
• *Let's check the position of the second hand. Is it at 36 seconds?*
• *The number 7 is at 35 seconds. The hand is pointing just past it. Yes, looks like 36 seconds.*
• *Wait, is the red hand really a second hand? Yes, it's thin and red.*
• *So the time is 12:36:36.*
• *Let me re-examine the minute hand. Maybe it's pointing at the 35 minute mark (the 7).*

[. . .]

• *This brings me back to the contradiction: Hour hand at 12, Minute hand at 35.*
• *If Minute hand is at 35, Hour hand should be halfway between 12 and 1.*
• *In the image, the hour hand is pointing very close to 12. Maybe slightly past.*

[. . .]

• *Let's assume standard clock: Black short = Hour, Black long = Minute, Red thin = Second.*

[. . .]

• *Let's assume the black hand is the minute hand. It is pointing at roughly the 36 minute mark.*
• *Let's assume the red hand is the second hand. It is pointing at roughly the 36 second mark.*
• *This would mean the time is roughly 12:36:36.*
• *Does the hour hand fit?*

[. . .]

- *Let's reconsider the hands.*
- *Maybe the hand pointing towards 7 is the HOUR hand?*
- *If Hour hand is at 7.*
- *Then Minute hand must be pointing towards... where?*
- *The other black hand is pointing towards 12.*
- *So Minute hand is at ~12 (0 minutes).*
- *So time is roughly 7:00.*

[. . .]

- *Let's assume the time is 12:36.*
- *Minute hand at 36 mins (just past 7).*
- *Hour hand at 12:36 position (halfway between 12 and 1).*
- *In the image, the hour hand is NOT halfway between 12 and 1. It*

[Generation ends here.]

Interpretation. The 2,048-token run identifies the red hand as a second hand but repeatedly revisits the black hands and remains uncertain about the second-hand value. Its extracted answer is `12:35`, obtained from the "So the time is" statement, even though it subsequently considers a conditional `7:00` interpretation. The longer run reaches `12:36:36`, including seconds and correcting the minute value, but continues reconsidering hand identities and the hour-hand position until the length limit. Its extracted answer comes from a `so_statement_hms` match, not from the final time mentioned in the trace. Thus, the correct extracted answer does not imply that re-evaluation has ceased. For this run, the reported 8,192-token total includes the prompt; the completion itself contains 7,344 tokens.

D Detailed ControlSet Evaluation Results

Figures 14 to 21 provide the complete matched pairwise comparisons underlying the ControlSet results in Section 4.2. Within each metric and model, each cell reports the accuracy of the row condition minus that of the column condition in percentage points. Coloured cells are statistically significant after the joint Benjamini–Hochberg correction described in Section 4.2 ($p_{\mathrm{adj}} < 0.05$); grey cells are not significant. For the second-hand skip rate, a negative difference means that the row condition produces fewer omissions; unlike the accuracy metrics, lower values are better.

D.1 Clock-Face Factors

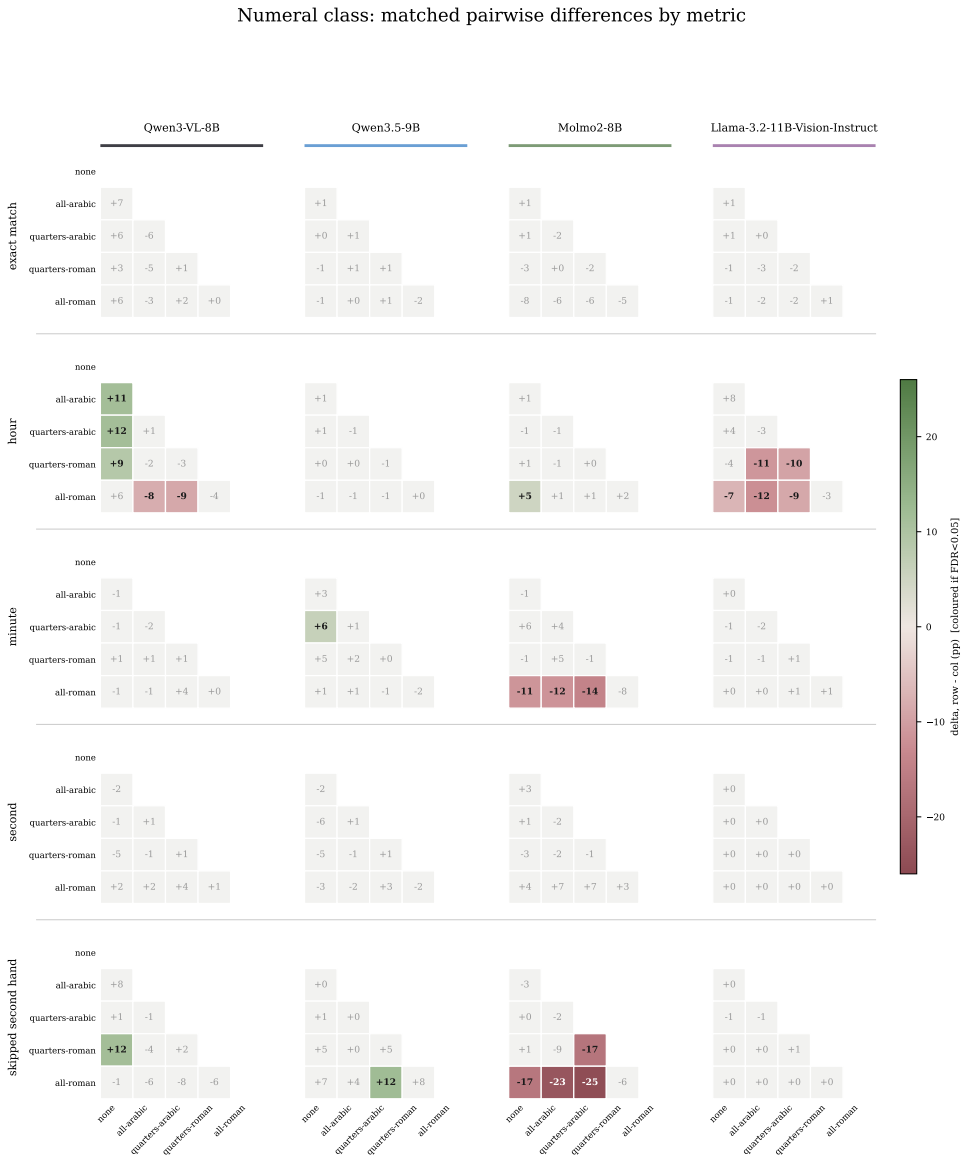


Figure 14: Matched pairwise differences between numeral-style conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.

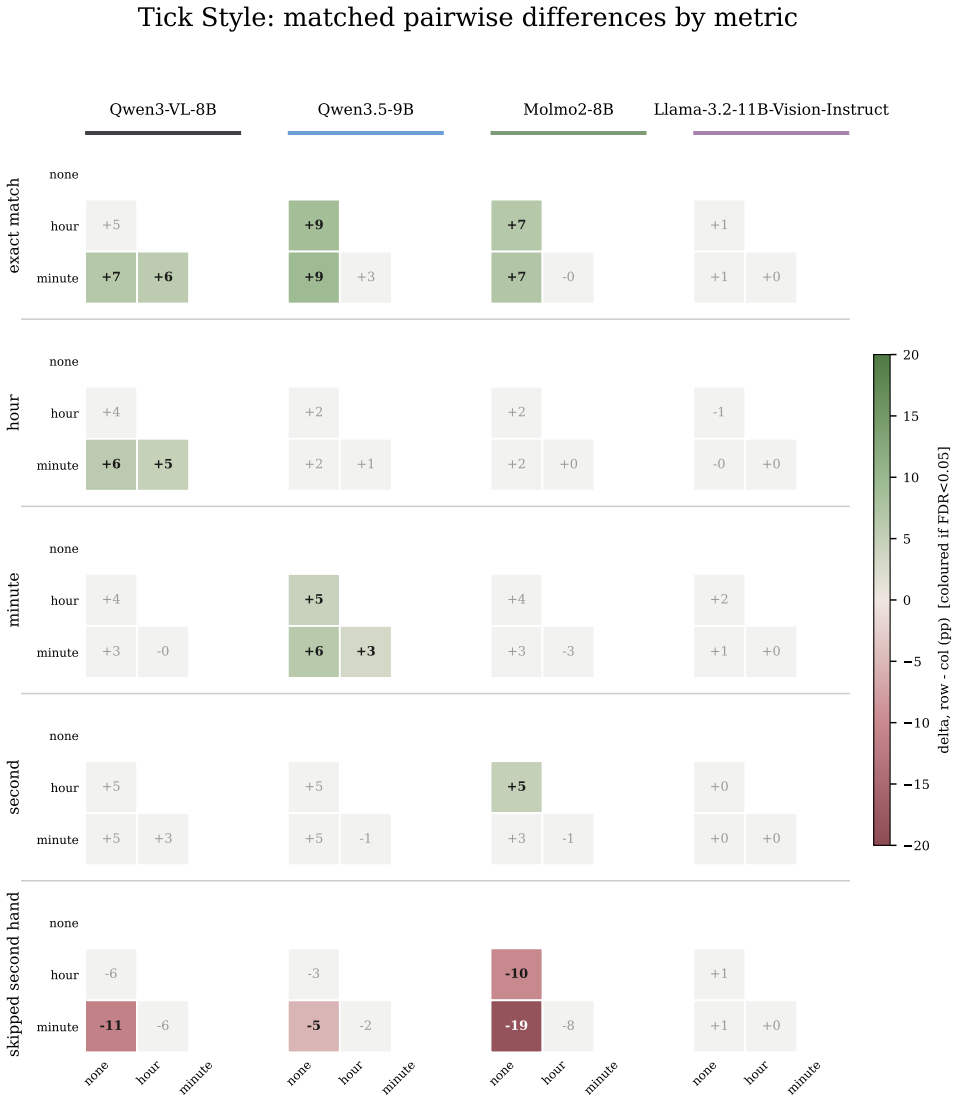


Figure 15: Matched pairwise differences between tick-style conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.



Figure 16: Matched pairwise differences between second-hand configurations for exact match, hour, minute, and second accuracy, and second-hand skip rate.

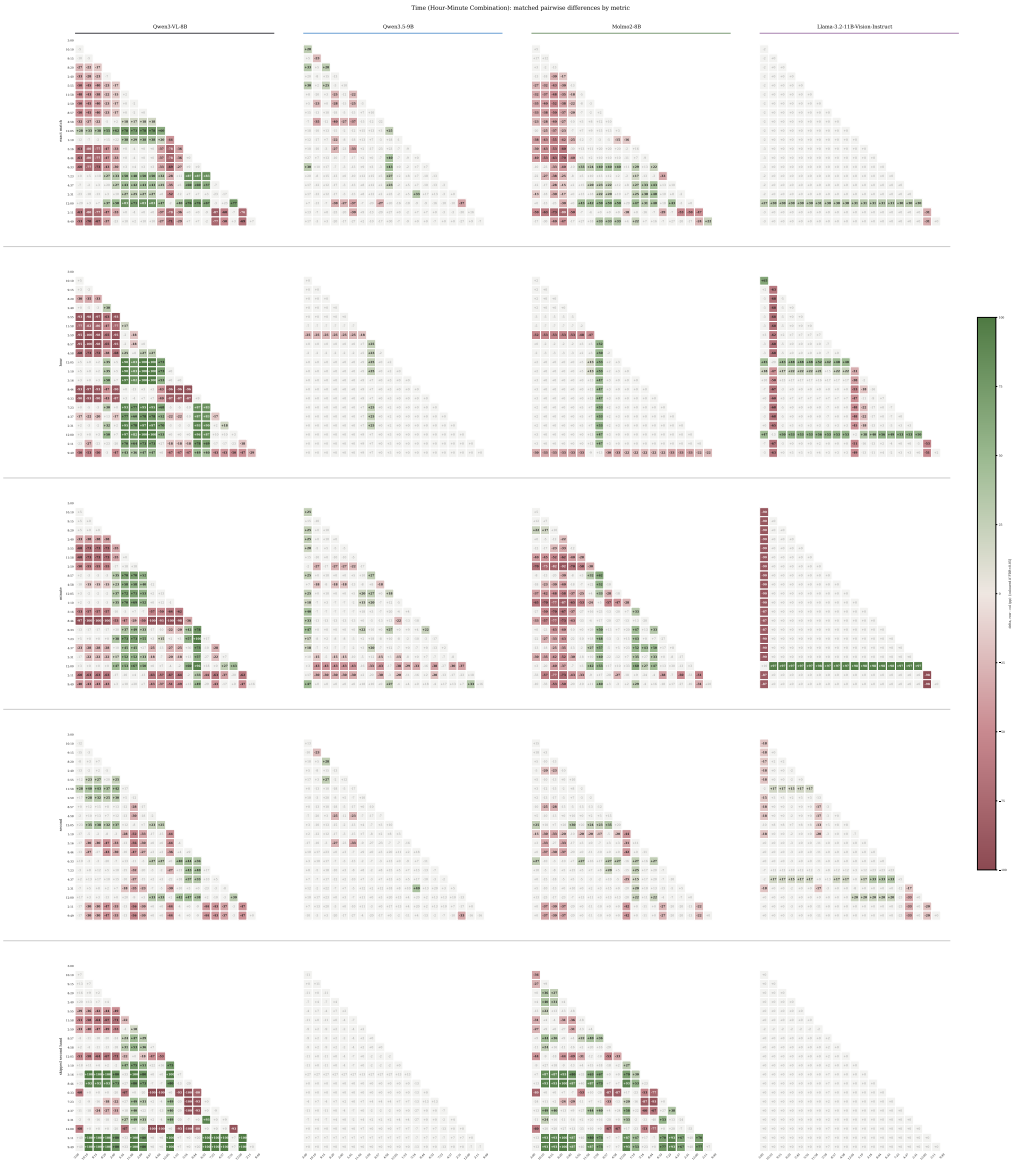


Figure 17: Matched pairwise differences among the selected hour–minute configurations for exact match, hour, minute, and second accuracy, and second-hand skip rate.

D.2 PRESENTATION FACTORS



Figure 18: Matched pairwise differences between canvas-size conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.

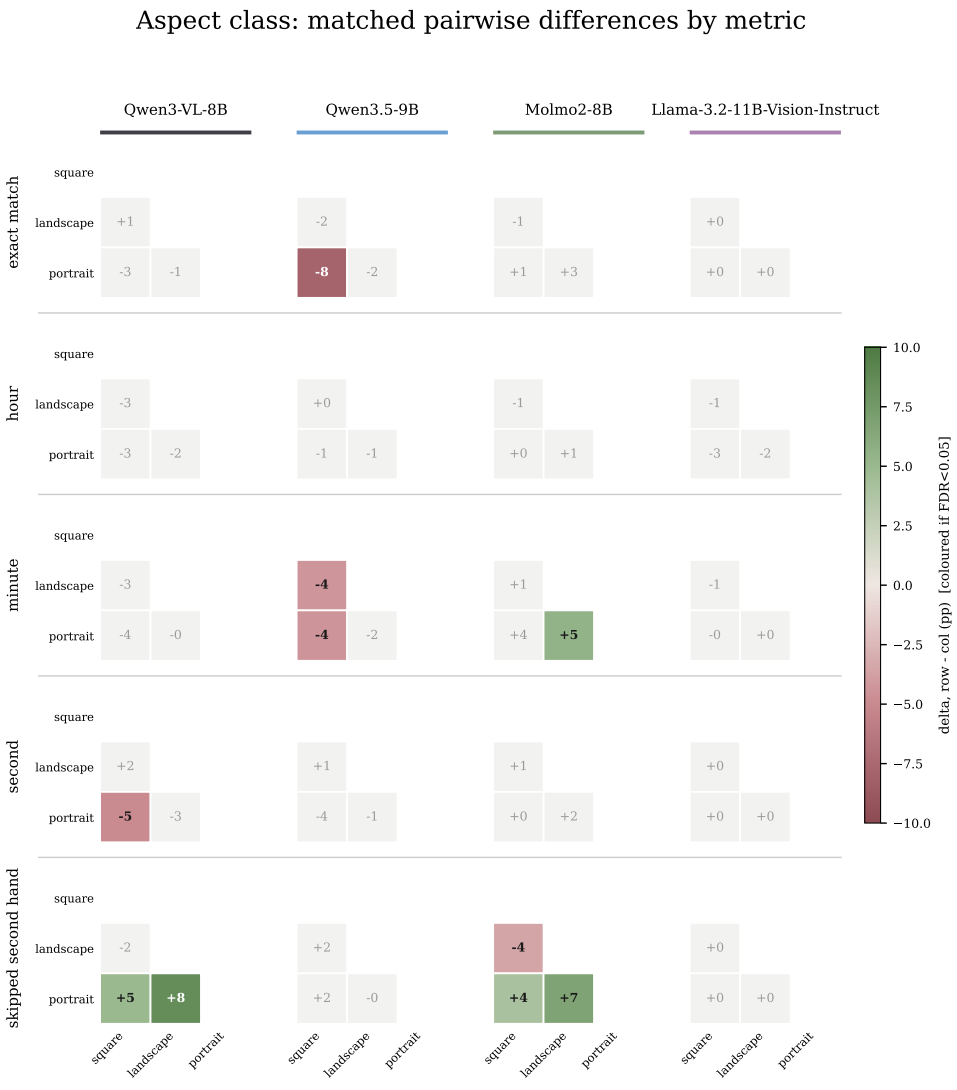


Figure 19: Matched pairwise differences between aspect-ratio conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.



Figure 20: Matched pairwise differences between clock-position conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.

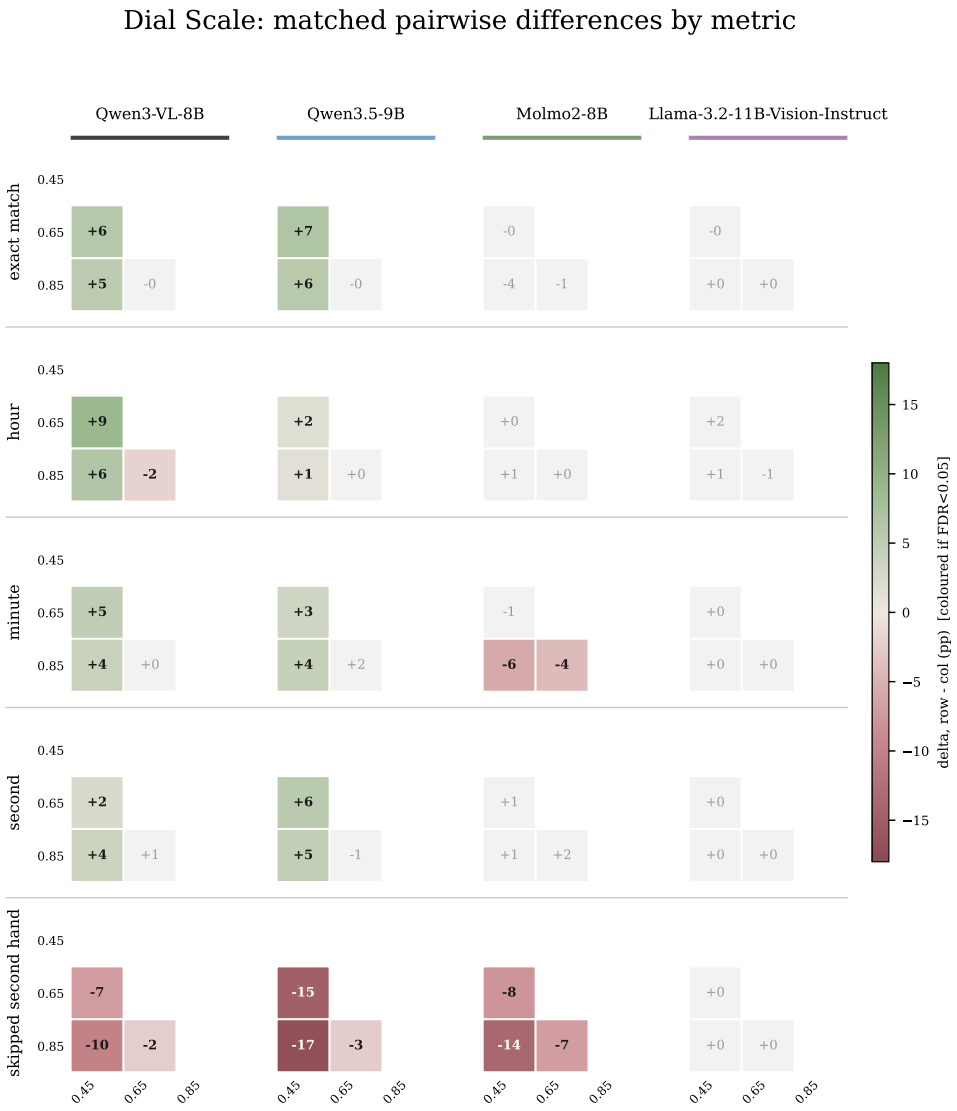


Figure 21: Matched pairwise differences between dial-scale conditions for exact match, hour, minute, and second accuracy, and second-hand skip rate.

D.3 PRE- AND POST-FINE-TUNING SENSITIVITIES

For an attribute with k possible values, the number of pairwise contrasts is $\binom{k}{2}$; for example, three tick styles yield three comparisons.

Table 18: Controlled sensitivities before and after fine-tuning. Each cell reports the percentage of pairwise exact-match contrasts that are significant after Benjamini–Hochberg correction ($p_{\mathrm{adj}} < 0.05$), as pre-SFT $\rightarrow$ post-SFT. Green marks an increase in sensitivity, red a decrease; unchanged values are greyed. Specific contrasts are reported in Appendix D.3.

	Clock attributes				**Presentation factors**			
Model	Numerals	Tick marks	Second hand	Displayed time	Aspect	Canvas size	Position	Dial scale
# pairs	10	3	10	253	3	21	36	3
Qwen3-VL	0.0 → 0.0	66.7 → 0.0	60.0 → 40.0	63.6 → 0.4	0.0 → 33.3	9.5 → 61.9	41.7 → 0.0	66.7 → 100.0
Qwen3.5	0.0 → 0.0	66.7 → 0.0	60.0 → 40.0	14.2 → 0.0	33.3 → 0.0	33.3 → 38.1	22.2 → 0.0	66.7 → 66.7
Molmo2	0.0 → 0.0	66.7 → 0.0	40.0 → 40.0	48.2 → 7.1	0.0 → 0.0	4.8 → 28.6	19.4 → 19.4	0.0 → 66.7
Llama	0.0 → 10.0	0.0 → 0.0	10.0 → 40.0	8.7 → 13.4	0.0 → 0.0	0.0 → 52.4	0.0 → 5.6	0.0 → 100.0

E Activation-Probing Protocol and Additional Results

E.1 Targets and Metric

We extract activations from every vision-encoder layer, the vision–language projector or multimodal interface, and a sparse set of layers spanning the language decoder. The exact sites are architecture-dependent and are identified along the horizontal axes of the result figures. Separate linear probes predict the physical angle of the hour, minute, and second hands. Predicting geometry rather than discrete clock values reduces the possibility that late-layer decodability merely reflects preparation of the answer tokens.

Because angles are cyclic, each target angle θ is represented by

$$\mathbf{y}_\theta = [\sin\theta, \cos\theta]^\top, \qquad \hat{\theta} = \operatorname{atan2}(\widehat{\sin\theta}, \widehat{\cos\theta}) \bmod 2\pi. \tag{1}$$

We report mean circular error (MCE), measured in degrees:

$$\mathrm{MCE} = \frac{1}{N}\sum_{i=1}^{N} \min(\Delta_i, 2\pi - \Delta_i)\,, \qquad \Delta_i = |\theta_i - \hat{\theta}_i|. \tag{2}$$

A prediction independent of the target has expected error $\pi/2 = 90^\circ$; lower MCE therefore indicates greater linear accessibility of hand orientation.

E.2 Data, Preprocessing, and Readout

For each experiment, we sample 1,000 training, 50 development, and 100 test examples from DiverseSynthetic. The same split sizes and procedure are used before and after SFT. These sets are smaller than the behavioural evaluation sets because probing requires retaining activations from many layers and visual tokens.

At sites containing spatial visual tokens, we apply adaptive pooling to obtain a fixed 8×8 grid. This retains coarse spatial structure while making the input dimensionality independent of the model's native visual-token resolution. Molmo's global and local-crop streams are probed separately; local-crop features are averaged across tiles before pooling where a common spatial representation is required.

The pooled activations are standardised and reduced to 256 channel-wise principal components at each spatial location, then flattened. The standardisation and PCA transforms are fitted only on the training split and applied unchanged to development and test examples. A ridge readout predicts the sine and cosine targets by minimising

$$\|Y - XW\|_F^2 + \lambda\|W\|_F^2, \qquad \lambda = 10. \tag{3}$$

The pooling resolution, PCA dimension, and regularisation strength were selected in preliminary sweeps and then fixed across models, layers, hands, and pre-/post-SFT comparisons. Consequently, differences in probe error reflect changes in linear accessibility under a common readout rather than layer-specific tuning of probe capacity.

E.3 Additional Layer-Wise Results

In Qwen3.5, SFT yields modest improvements within the vision encoder but sharply reduces error after projection and through the language decoder. At the merger, the reductions are approximately 17°, 23°, and 12° for the hour, minute, and second hands, respectively. This stronger projector effect than in Qwen3-VL is consistent with Qwen3.5 relying on a single projected visual stream rather than Qwen3-VL's additional DeepStack injections. Decodability alone does not determine behaviour: before SFT, Qwen3-VL attains a lower best second-hand error than Qwen3.5 (approximately 29° versus 39°), despite lower second-component accuracy on the DiverseSynthetic `H:MM:SS` subset (36.8% versus 83.5%).

For Molmo2, the global view is generally more informative in the vision encoder and early decoder, whereas the reassembled high-resolution stream becomes especially useful for the second hand after

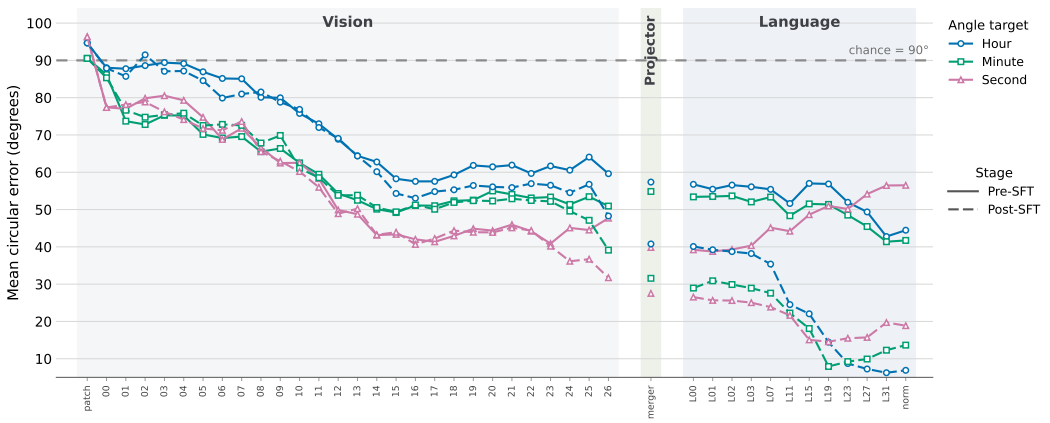


Figure 22: Layer-wise hand-angle probe error for Qwen3.5-9B before and after joint vision–projector–language fine-tuning. Lower is better; the black dashed line marks the 90° random baseline. Improvements are largest at the projector and throughout the language decoder.

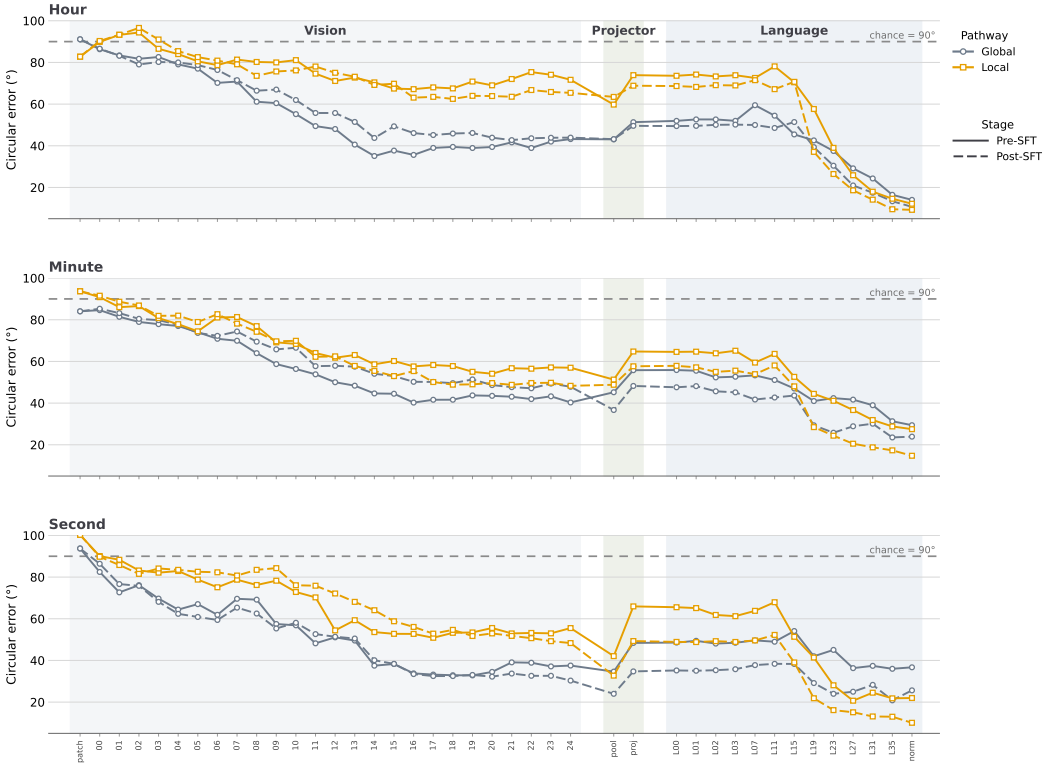


Figure 23: Layer-wise hand-angle probe error for Molmo2-8B before and after vision-only fine-tuning, with global and reassembled high-resolution streams shown separately. Lower is better; the black dashed line marks the 90° random baseline. Although error rises at the projector, hand geometry—especially the second-hand angle—recovers in the language decoder.

the local tiles are restored to spatial order. Angle decodability initially degrades at the attention-pooling projector but recovers from approximately the middle of the decoder, suggesting a change of representational basis rather than irreversible information loss. Vision-only SFT produces its

largest reductions in minute- and second-hand error downstream in the unchanged language decoder, showing how an adapted visual pathway can reorganise later activations without updating decoder weights.

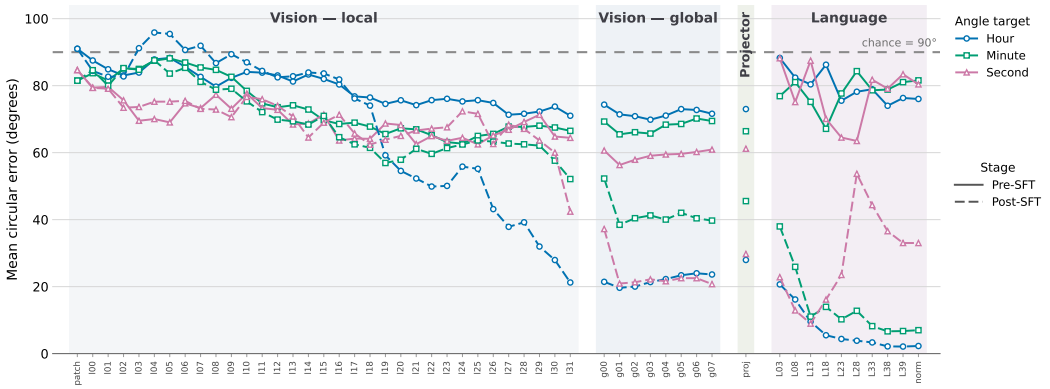


Figure 24: Layer-wise hand-angle probe error for Llama-3.2-11B-Vision-Instruct before and after joint vision–language fine-tuning. Lower is better; the black dashed line marks the 90° random baseline. Fine-tuning improves both pathways and produces precise hour- and minute-hand representations in the decoder.

Llama has the weakest base representations: vision-layer errors are typically 60–70°, and decoder errors remain near chance after visual information first enters through cross-attention at layer 3. Joint vision–language SFT changes both pathways. In the global vision transformer, the best hour, minute, and second errors fall from approximately 70°, 65°, and 56° to 20°, 38°, and 21°, respectively. In the decoder, hour and minute errors reach 2.1° and 6.7° near the output. Second-hand error reaches 9.0° at an intermediate layer but rises to roughly 33° at the end, reproducing the late loss of fine-grained angle information observed in the Qwen models. These changes accord with the behavioural finding that Llama benefits most from the broader vision–language intervention.

E.4 Hand-Wise Decodability and Accuracy Gains

Figure 25 complements the aggregate analysis in Figure 4 by separating the relationship between decodability gains and clock-reading accuracy gains for the hour, minute, and second hands. The plotting procedure and layer-wise summary are otherwise unchanged.

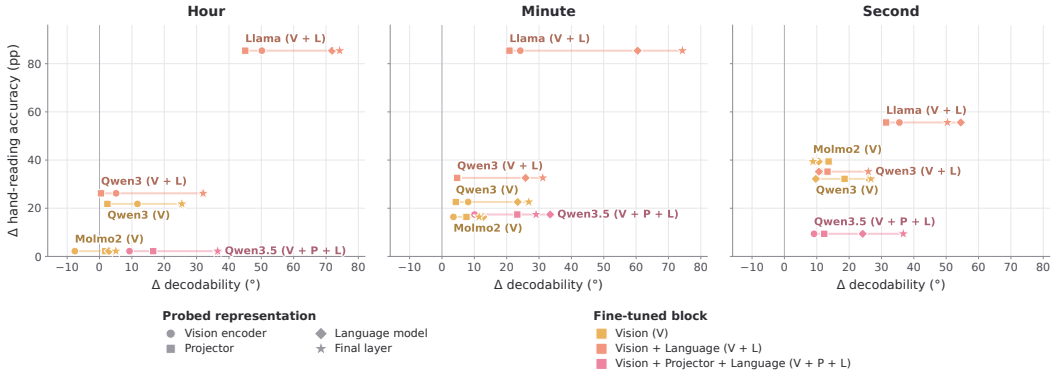


Figure 25: Change in clock-reading accuracy on the DiverseSynthetic test set versus change in hand-angle decodability after fine-tuning, separated by the hour, minute, and second hands. For each hand and representation block, decodability is summarised by the lowest mean circular error across its probed layers. Accuracy and decodability changes are measured relative to the corresponding base model, following the same procedure as in Figure 4.

F CAUSAL ACTIVATION-PATCHING PROTOCOL

For each CausalClock pair, the base and counterfactual images share the same rendering and differ in exactly one hand position (Appendix B.4). We first run the model independently on both images. We then repeat the base-image run while replacing the activation at one network site with the corresponding activation from the counterfactual run. Within the language model, we separately patch the vision-token positions, all textual prompt positions, or only the final prompt token. This procedure tests whether the selected activation is sufficient to move the model from the base reading toward the matched counterfactual reading.

For each run, we measure the model's relative preference for the counterfactual answer over the base answer using the log-probability margin

$$M = \mathrm{meanlogP}(y_{\mathrm{cf}}) - \mathrm{meanlogP}(y_{\mathrm{base}}), \tag{4}$$

where y_{cf} and y_{base} are the counterfactual and base answer-token sequences. Averaging over answer tokens avoids making the margin depend primarily on answer length. We define the fraction of the counterfactual endpoint gap recovered as

$$R = \frac{M_{\mathrm{patched}} - M_{\mathrm{base}}}{M_{\mathrm{cf}} - M_{\mathrm{base}}}. \tag{5}$$

The denominator is the complete change caused by replacing the base image with the counterfactual image; the numerator is the change caused by patching a single activation site. The resulting values have a direct interpretation:

- $R = 0$: patching causes no movement from the base-image behaviour;
- $R = 0.5$: patching recovers half of the counterfactual-image effect;
- $R = 1$: patching reaches the full counterfactual endpoint;
- $R > 1$: patching moves beyond the counterfactual endpoint; and
- $R < 0$: patching moves in the opposite direction.

When the base and counterfactual margins are effectively identical, the denominator contains no meaningful endpoint gap; the recovery value is therefore left undefined and omitted.

For example, suppose the base, counterfactual, and patched margins are -4, 6, and 1, respectively. Then

$$R = \frac{1 - (-4)}{6 - (-4)} = 0.5, \tag{6}$$

so the patch recovers 50% of the effect produced by the full counterfactual image. This probability-based metric is more sensitive than exact-answer flip rate: a patch may strongly shift the relative answer probabilities without changing which answer is ultimately generated.

The curves in Figure 3 report mean recovery across examples. Shaded regions show 95% confidence intervals computed as ± 1.96 cluster-robust standard errors.

Figure 26 further stratifies the visual-token intervention by the magnitude of the counterfactual hand displacement. Although recovery differs somewhat across perturbation scales in the vision encoder, the trajectories become similar after visual information enters the language decoder. The other token scopes exhibit the same qualitative relationship with displacement scale and are omitted for clarity.

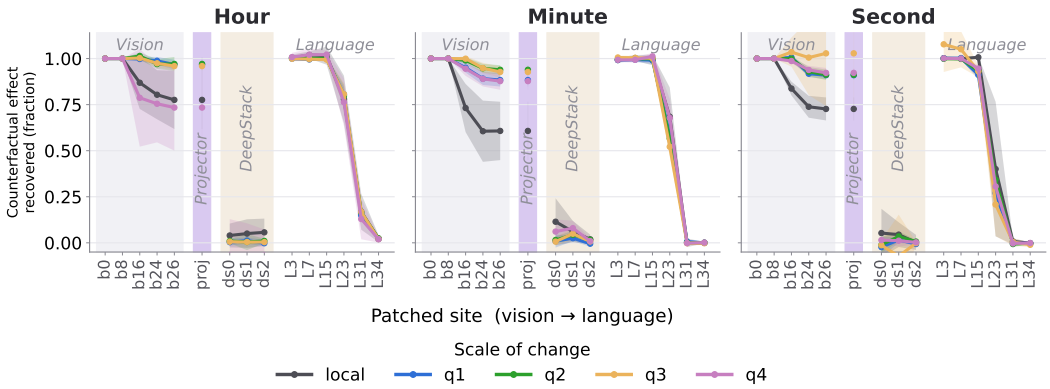


Figure 26: Fraction of the counterfactual effect recovered across Qwen3-VL-8B-Instruct depth, stratified by the magnitude of the counterfactual hand displacement. Panels correspond to the perturbed hand, and lines denote the displacement bands defined in Table 11; only vision-token patches are shown. Recovery varies somewhat across displacement scales in the vision encoder, but the trajectories largely converge in the language decoder.

G TARGETED FINE-TUNING PROTOCOL

G.1 TRAINING SETUP

All interventions use supervised fine-tuning with LoRA while the pretrained parameters outside the selected adapter targets remain frozen. Depending on the intervention, adapters are applied to the vision encoder, multimodal projector or merger, language decoder, or combinations of these components. The common configuration uses rank $r = 16$, scaling parameter $\alpha = 32$, dropout 0.05, and no bias training. The resulting trainable parameter counts are reported in Table 6.

Unless stated otherwise, training runs for at most 1,500 optimiser updates on a single NVIDIA A100 40GB GPU. We use AdamW (Loshchilov & Hutter, 2019) with a learning rate of 2×10^{-4}, linear decay, a warm-up ratio of 0.03 (approximately 45 updates), and no weight decay. The per-device batch size is 2 with 2 gradient-accumulation steps, giving an effective batch size of 4. Training uses bfloat16 precision, non-reentrant gradient checkpointing, a maximum sequence length of 8,192 tokens, and random seed 0.

The Llama interventions run for at most 3,000 optimiser updates. Because a per-device batch size of 2 exceeds the available memory, we use a batch size of 1 and 4 gradient-accumulation steps, preserving the effective batch size of 4. All other settings remain unchanged.

G.2 VALIDATION AND CHECKPOINT SELECTION

We validate and save checkpoints every 50 optimiser updates. Validation uses greedy decoding with at most 128 newly generated tokens. Each saved checkpoint is first evaluated on a fixed 50-example subset of the 300-example DiverseSynthetic development set. We then evaluate the three checkpoints with the highest subset exact-match accuracy on the complete development set and select the checkpoint with the highest full-set exact-match accuracy for the reported test results.

G.3 Complete Intervention Results

Table 19: Exact-match accuracy for all LoRA intervention presets on the in-distribution DiverseSynthetic test set and four out-of-distribution benchmarks. MB Synth. and MB Real denote the synthetic and real-world MeasureBench subsets; TTVQA denotes TickTockVQA-Segmented. Grey marks the baseline, green highlights the preset with the highest in-distribution accuracy for each model, and bold marks the best result in each evaluation column within each model.

Intervention	LoRA share	ID test	DateTimeQA	MB Synth.	MB Real	TTVQA
GPT-5.5	–	45.80	30.65	47.41	32.20	14.40
GPT-5.5 (high reasoning effort)	–	59.20	41.94	50.37	40.68	16.20
Qwen3-VL						
Baseline (no SFT)	–	23.60	53.23	21.11	42.37	18.40
Vision	0.088%	70.40	66.13	64.81	41.53	19.00
Projector	0.013%	32.00	46.77	34.07	33.05	19.80
Projector (MLP only)	0.003%	57.80	58.06	54.81	31.36	21.00
Language	0.498%	79.40	85.48	74.07	46.61	27.60
Projector + language	0.511%	80.00	77.42	75.19	46.61	26.00
Vision + language	0.586%	**84.80**	72.58	**78.15**	**48.31**	27.60
Vision + projector + language	0.599%	83.20	**87.10**	**78.15**	43.22	**28.80**
Qwen3.5						
Baseline (no SFT)	–	69.00	88.71	80.37	46.61	38.20
Vision	0.082%	90.00	87.10	84.44	58.47	37.60
Projector	0.003%	89.20	88.71	84.44	55.08	36.80
Language	0.460%	91.00	90.32	81.48	67.80	38.60
Language (full attention)	0.042%	90.60	**93.55**	82.59	**68.64**	38.20
Projector + language	0.463%	90.80	90.32	81.48	65.25	37.80
Vision + language	0.542%	92.40	90.32	85.93	67.80	**41.80**
Vision + projector + language	0.545%	**92.80**	83.87	**92.80**	63.56	36.60
Molmo2						
Baseline (no SFT)	–	29.40	43.55	47.78	44.92	33.00
Vision	0.093%	**71.00**	**70.97**	**78.89**	**56.78**	38.20
Projector	0.010%	55.60	59.68	75.56	**56.78**	37.80
Projector (MLP only)	0.008%	56.40	59.68	75.56	**56.78**	37.80
Language	0.422%	69.00	67.74	77.04	51.69	42.20
Projector + language	0.433%	68.60	59.68	52.96	19.49	44.20
Vision + language	0.515%	50.40	62.90	54.81	22.88	**47.60**
Vision + projector + language	0.525%	50.80	59.68	52.96	19.49	44.20
Llama						
Baseline (no SFT)	–	0.00	1.61	0.00	0.85	0.20
Vision	0.138%	53.00	53.00	34.07	11.86	5.80
Projector	0.002%	0.20	12.90	0.74	3.39	1.20
Language	0.491%	63.40	43.55	54.81	**27.66**	10.60
Language (self-attention only)	0.459%	49.40	45.16	31.11	9.32	9.40
Language (cross-attention only)	0.032%	20.20	19.35	11.48	6.78	2.40
Projector + cross-attention	0.034%	25.40	22.58	20.37	8.47	3.40
Projector + language	0.493%	62.60	54.84	42.96	11.02	13.20
Vision + language	0.630%	**85.40**	**58.06**	**57.04**	27.12	**18.00**
Vision + projector + language	0.631%	85.00	**58.06**	54.44	26.27	14.20

H GENERALISATION BEYOND CLOCKS

H.1 MEASUREBENCH DIAL-INSTRUMENT EXAMPLES

Table 20 shows representative synthetic and real-world examples of the eight non-clock dial instruments used in the transfer evaluation.

Table 20: Synthetic and real-world examples of the non-clock dial instruments in MeasureBench. Dashes indicate subsets with no samples.

Instrument	**Synthetic example**	**Real-world example**
Ammeter		
Compass		
Fuel gauge	–	
Hygrometer		
Pressure gauge		
Speedometer	–	
Voltmeter		–
Weighing scale		–

H.2 CHECKPOINT-LEVEL TRANSFER TO NON-CLOCK INSTRUMENTS

Figure 27 compares clock-reading accuracy with accuracy on each non-clock dial instrument for every evaluated checkpoint. Improvements on DiverseSynthetic do not generally predict improvements on the corresponding MeasureBench instrument subsets.

H.3 INSTRUMENT-ATTRIBUTE DECODABILITY

Figure 28 extends the probing analysis to non-clock instruments. Needle angle is highly accessible for pressure gauges and ammeters, while the ammeter's joint unit-and-range category is almost perfectly linearly decodable early in the vision encoder and remains so after clock fine-tuning.

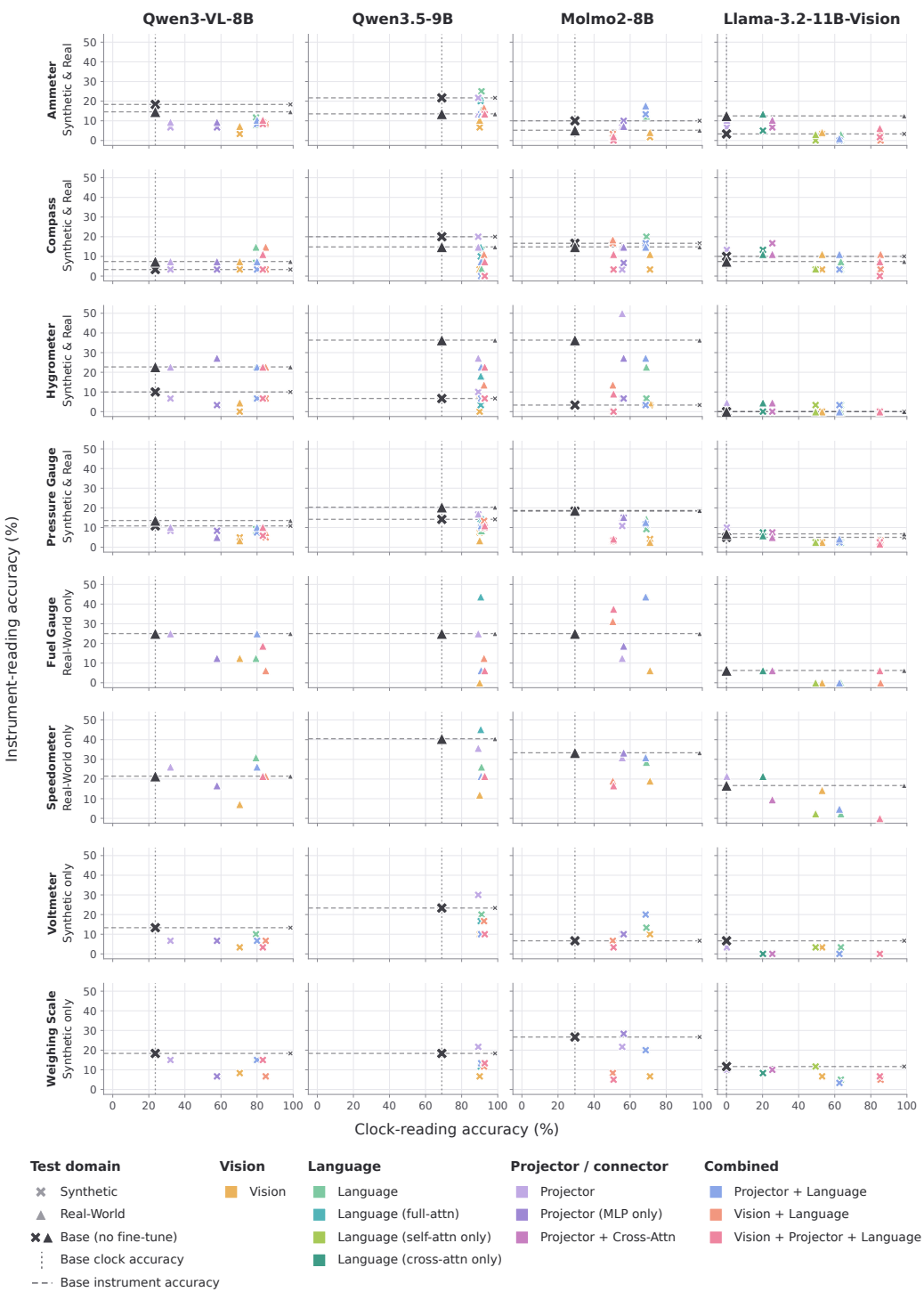


Figure 27: Clock-reading accuracy versus performance on non-clock dial instruments. Columns correspond to models and rows to instruments; each point represents one checkpoint. The x-axis reports clock-reading accuracy on the DiverseSynthetic test set, and the y-axis reports accuracy on the corresponding MeasureBench instrument subset. Both accuracies are percentages.

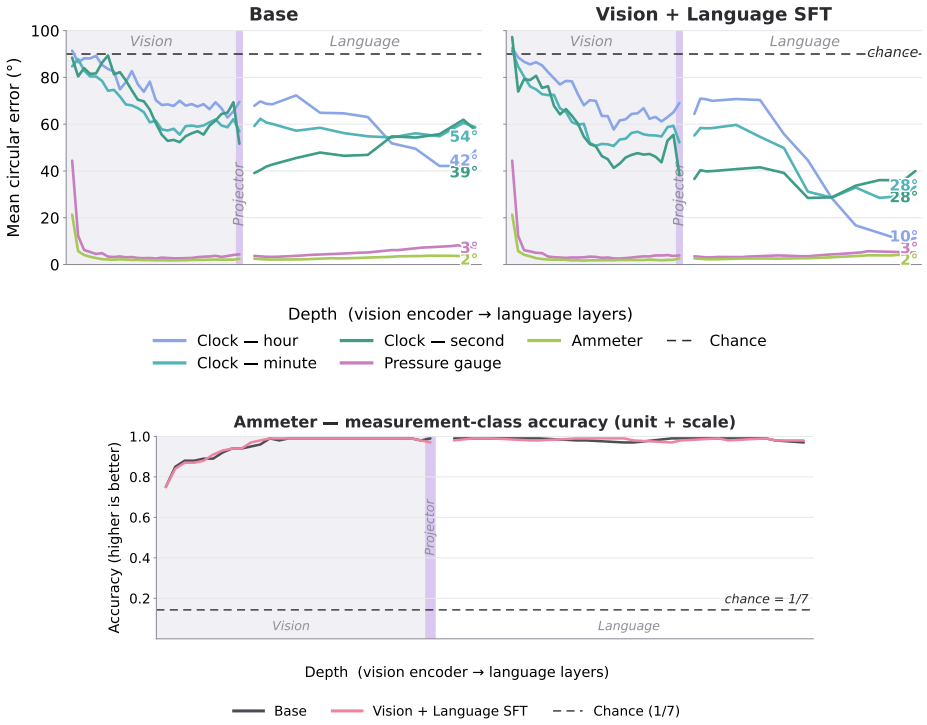
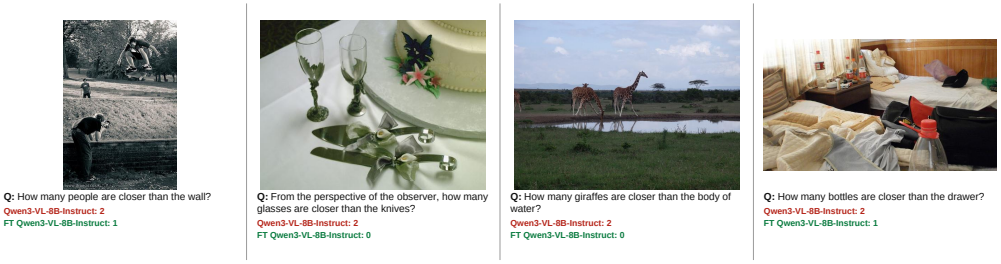


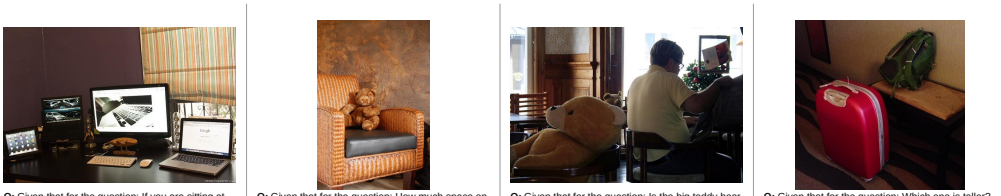
Figure 28: Instrument-attribute probes for the base Qwen3-VL-8B-Instruct and its vision–language fine-tuned checkpoint. *Top:* Layer-wise angle error for three clock hands and the needles of two single-pointer instruments; lower is better. Gauge needles reach approximately 2–3° error in the vision encoder. *Bottom:* Linear-probe accuracy for the ammeter's joint unit-and-range category (7 classes; chance $= 1/7$), which reaches near-ceiling accuracy early in the vision encoder and remains high after fine-tuning.

H.4 Qualitative Spatial-Reasoning Examples

The aggregate SPHERE results conceal heterogeneous changes across examples and spatial skills. Figures 29 and 30 show representative predictions that respectively improve and degrade after clock-reading fine-tuning.

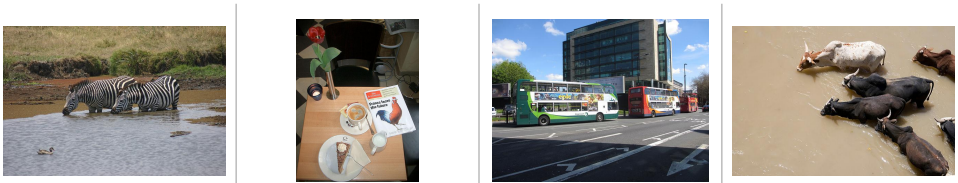


(a) Improved Multi-Skill: distance and counting

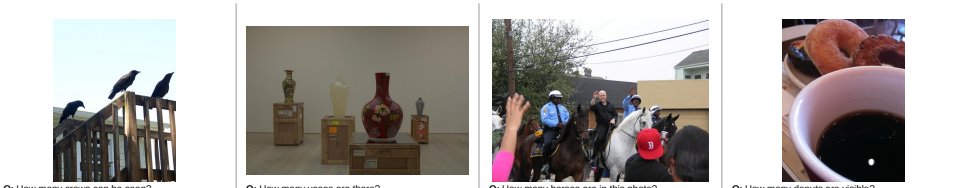


(b) Improved Reasoning: object manipulation with intermediate states

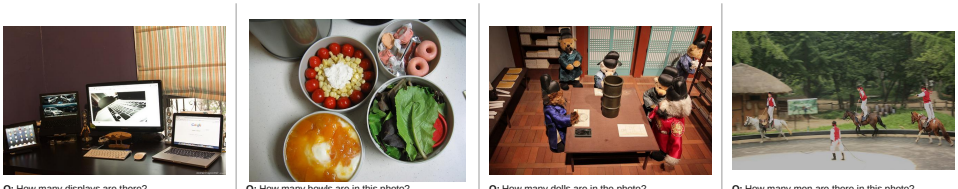
Figure 29: Qualitative SPHERE examples for which the selected Qwen3-VL checkpoint changes an incorrect baseline prediction to a correct prediction after synthetic clock-reading fine-tuning.



(a) Degraded Multi-Skill: position and counting



(b) Degraded Single-Skill: counting paired with distance and counting



(c) Degraded Single-Skill: counting paired with position and counting

Figure 30: Qualitative SPHERE examples for which synthetic clock-reading fine-tuning changes a correct baseline prediction to an incorrect prediction. Degradation is especially apparent in these counting examples; aggregate results are reported in Table 5.