REALNLR: DO VLMS LEARN NONLOCAL VISUAL ALGORITHMS FROM SYNTHETIC FINE-TUNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) do well on standard benchmarks but fail simple tasks that require chaining visual evidence across an image, called nonlocal visual reasoning (NLR): re-identifying a transformed object, following a chain of cues (saccadic search), and tracing a wire through clutter (contour tracing). We ask whether fine-tuning can teach these visual algorithms, and what the model learns if it does. We fine-tune five open VLMs (4B to 32B) with QLoRA on synthetic versions of the three tasks. We vary the adapted side (language or vision), the supervision (answer only, or intermediate steps too), the amount of data, and the rendering style (default or randomized). To measure transfer, we introduce RealNLR[1], a benchmark that poses the same tasks on realistic images. We find that (i) fine-tuning raises in-distribution accuracy by 13 points on average, and the best runs go from near zero to 98%, but transfer to RealNLR is zero; (ii) process supervision teaches the steps, with InternVL3-14B writing 96% of out-of-distribution search trajectories exactly, yet its answer still stops at the chain lengths seen in training; (iii) internal analyses show that fine-tuning improves one stage of the procedure on synthetic images, but not how the model builds visual state from a natural image; and (iv) randomized-style training does not improve transfer (mean change -0.6 points against a matched control). The models learn the dataset, not the algorithm.

1 INTRODUCTION

Vision-language models (VLMs) answer questions about charts and natural scenes with high accuracy, yet fail visual tasks that humans find easy: deciding whether a shape from one panel reappears in another, following a chain of written cues, or tracing one wire through a tangle of others. Berman & Deng (2025) group these tasks under nonlocal visual reasoning (NLR), where the answer depends on evidence from several regions of the image, and show that frontier models score near zero on at least one variant of each task.

It is not clear whether training can close this gap. Fine-tuning on visual-cognition tasks improves the trained task but rarely transfers (Schulze Buschoff et al., 2025). Synthetic data has produced real-image gains on single-step localization (Rizzoli et al., 2025). And synthetic accuracy alone is weak evidence, because a model can learn shortcut cues instead of the intended procedure (Geirhos et al., 2020). We ask three questions. Can these visual algorithms be learned from synthetic data? Where in the model does the learning happen? When a fine-tuned model succeeds, has it learned the procedure or a shortcut?

We fine-tune five open VLMs (4B to 32B) with QLoRA (Dettmers et al., 2023) on the three tasks of Berman & Deng (2025). We vary which side of the model is adapted (language or vision, with a parameter-matched control), the supervision format (answer only, or intermediate steps and answer), and the amount of data. We test on in-distribution puzzles, a harder out-of-distribution split, and RealNLR, a benchmark we build that poses the same tasks on realistic images. We add two controls. We retrain Qwen3-VL-8B on renderings with randomized visual style, and we repeat its default-style runs with a second seed.

[1]Anonymous benchmark release: `https://anonymous.4open.science/r/RealNLR/`

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.

To interpret the results, we split nonlocal reasoning into three stages: construct the relevant state from the image, operate on that state, and convert the result into an answer. In our analyses (§5), fine-tuning changes the second or third stage on synthetic images. It does not improve the first stage on natural images, which is where RealNLR fails.

Our contributions are: **(i)** a controlled fine-tuning matrix over three tasks, five VLMs, three adaptation regimes, two supervision formats, and three data scales; **(ii)** RealNLR, a benchmark with realistic analogs of those tasks; **(iii)** evidence that synthetic gains do not transfer, and that process supervision teaches the steps without changing how the answer is computed; **(iv)** a randomized-style training study and a second-seed replication, which show that randomizing rendering style does not change transfer and measure how much single runs vary; and **(v)** an analysis that locates what fine-tuning changes within the three stages.

2 Related Work

Nonlocal visual reasoning. VLMs with strong VQA and chart scores still fail perceptual tests that are trivial for humans: counting line intersections (Rahmanzadehgervi et al., 2024), telling apart image pairs (Tong et al., 2024), and reading angles and sizes (Kamoi et al., 2025). Most of these tests can be solved by reading facts off the image one at a time and reasoning in text. Berman & Deng (2025) describe the case where that does not work, nonlocal visual reasoning, through three task families in which evidence must be chained across the image: comparing two objects held in memory, jumping between cues, and following a contour. These works evaluate and diagnose the problem, but none train on it. We use the generators of Berman & Deng (2025) as training data and test on realistic analogs in RealNLR, so that a gain on real images can only come from the trained skill.

Fine-tuning on synthetic visual data. Several studies fine-tune VLMs on narrow synthetic tasks such as intuitive physics (Schulze Buschoff et al., 2025) or object localization (Rizzoli et al., 2025) and evaluate on real images such as COCO. Others generate synthetic or verifiable perception tasks and report gains on standard benchmarks (Wang et al., 2025; Zhou et al., 2026). These studies change the data and ask whether scores improve. Gains on synthetic tasks can come from shortcut cues (Geirhos et al., 2020), and SFT tends to memorize its training distribution where reinforcement learning generalizes (Chu et al., 2025). On the supervision side, step-level feedback helps multi-step reasoning in LLMs (Lightman et al., 2023), and visual chain-of-thought work supplies intermediate regions and reasoning steps (Shao et al., 2024). We fix the tasks and vary the training signal: whether supervision shows the answer or the steps, how much data is used, and how the images are rendered. Process targets come from the generator's scene graph and are scored step by step. We balance answers and test on a harder split so that a shortcut cannot explain a score increase.

Where perception fails inside a VLM. Some failures happen after visual encoding: the needed information is present in visual or internal representations but is not reliably decoded or used when the model answers (Huang et al., 2025; Rahmanzadehgervi et al., 2024; Shahgir et al., 2026). Visual path-following failures have also been traced to local competition from nearby distractors (Hong et al., 2026). That evidence comes from probes and behavior, and the fine-tuning studies above adapt the whole model, so neither says which side has to change. We test it by intervention, adapting the vision side and the language side separately with matched parameter budgets.

3 Methodology

3.1 Synthetic Tasks

We train on the three synthetic NLR tasks of Tunnel Vision (Berman & Deng, 2025), using and extending its procedural generators for Object Re-Identification, Visual Scavenger Hunt (our saccadic search), and Circuit Connections (our contour tracing). Appendix A gives the prompts. We generate 5,000 training examples per task with balanced answers. Chance is 50% for object re-identification and 12.5% for saccadic search and contour tracing. Every queried wire crosses at least one other wire, since sensitivity to crossings separates tracing from guessing by endpoint proxim-

ity (Berman & Deng, 2025). We render each circuit three ways (Figure 15): all wires one color, every wire a different color, and black wires with small labeled markers for process SFT.

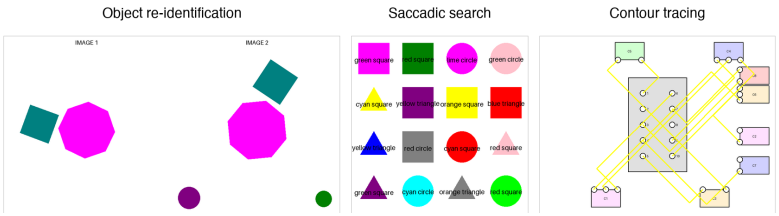


Figure 1: One example per task: object re-identification, saccadic search, and contour tracing.

Each task has two held-out test sets of 150 examples. The in-distribution (ID) set uses the training generator settings. The out-of-distribution (OOD) set raises them: more parts and distractors for re-identification, a 5×5 grid and longer chains (up to 6 hops, against at most 4 in training) for saccadic search, and more wires with at least two crossings for circuits. ID accuracy tells us whether the model learned the task. The ID-to-OOD gap tells us whether it learned the procedure or only the training distribution.

Randomized style. A model trained on one rendering can fail on real images for reasons unrelated to the algorithm. To test this, we render the same tasks in randomized style. The generator samples the same shapes, chains, wires, and answers, and a separate random stream then draws, per image, the background (tinted or dark, with optional texture and pixel noise), the shade of each named color, the font and its size, line and outline widths, the panel layout, cell and canvas sizes, shape outlines, grid lines, and label placement and case (Figure 16). Task logic, prompts, and answers do not change. We build randomized training sets of 5,000 examples per task and randomized ID and OOD test sets of 150 each.

3.2 RealNLR Construction

RealNLR poses the same three tasks on realistic images. We built it around three rules: take ground truth from existing object annotations; measure chance accuracy per task and minimize shortcuts; and crop each example so a human can read the prompt-relevant region. Appendix B describes each family, Table 1 lists the filtering and rendering parameters, and Figure 11 gives human accuracy.



Figure 2: RealNLR examples. Object re-identification (left) matches a reference crop to a boxed candidate. Saccadic search (center) follows object labels through a hop chain. Contour tracing (right) asks which green candidate lies on the contour of the cyan reference point.

3.3 Fine-tuning

We fine-tune Gemma 3 4B IT (Gemma Team et al., 2025), Qwen3-VL 4B Instruct (Bai et al., 2025), Qwen3-VL 8B Instruct, InternVL3 14B (Zhu et al., 2025), and Qwen3-VL 32B Instruct on all

three tasks, for 3 epochs at 100, 1,000, and 5,000 examples, with seed 808. There are two main regimes: language-side (LoRA on the language model, vision frozen) and vision-side (LoRA on the vision encoder, language frozen). The language side is larger, so it trains more parameters. As a control, Qwen3-VL 8B also gets a vision-side run with the same number of trainable parameters as its language-side run (matched vision-side). All five models train on the answer alone (answer SFT). Qwen3-VL 8B and InternVL3 14B also train to write the intermediate steps (process SFT) for saccadic search and contour tracing; object re-identification is a single comparison with no steps to supervise. The process target is a JSON object with sorted keys, so the answer field comes *before* the trajectory or marker list. The model therefore commits to its answer before it writes any step. Process SFT supervises the steps as an extra output, not as reasoning that precedes the answer. Table 2 lists trainable parameters and Appendix C the hyperparameters.

For the style study we train Qwen3-VL 8B at n=5,000 on the randomized renderings, for all five task and supervision pairs, under the language-side and matched vision-side regimes and a connector-only regime that adapts only the vision-to-language merger (1.1M parameters), with seeds 808 and 809. The randomized saccadic process target also records the grid row and column of each visited cell. As the control, we repeat the default-style Qwen3-VL 8B runs at n=5,000 with seed 809.

3.4 Testing

We evaluate the base and fine-tuned models on the synthetic ID and OOD sets and on RealNLR, after each epoch. Decoding is greedy. Answer-only generation stops after 8 tokens and process generation after 1,000. For process SFT we score the full output, the trajectory (or marker sequence) alone, and the answer alone. The main matrix uses 300 random RealNLR questions per task. The style study scores the full RealNLR sets of the benchmark release (877 re-identification, 549 saccadic, and 1,000 contour examples). Its accuracies are not directly comparable to the 300-question numbers of the main matrix. Re-identification in particular scores 70 to 82% on the full set, against 35 to 63% on the 300 questions, and the release uses a later revision of the re-identification files. Every randomized run has a default-style control scored on the same examples. Figure 14 (Appendix H) summarizes the pipeline.

4 Results

Fine-tuning helps in distribution, less out of distribution. Across 159 epoch-3 conditions (answer SFT, process SFT scored on the answer, and process SFT scored on the full output), fine-tuning raised ID accuracy by 13.37 points on average and OOD accuracy by 6.98 points. The ID median gain was only 6.0 points while the best run reached 100%, so a few conditions carry the mean. On ID, 129 of 159 conditions improve, 22 get worse, and 8 tie; on OOD, 119 improve, 21 get worse, and 19 tie. Figure 12 (Appendix E) shows every condition.

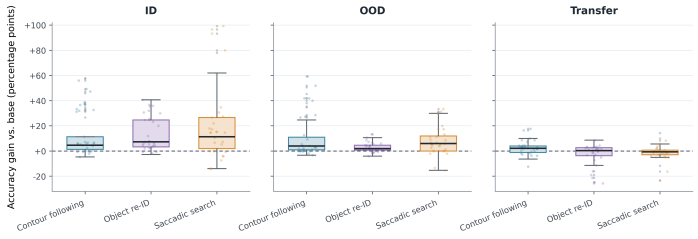


Figure 3: Accuracy gain over the base model at epoch 3, per task, on synthetic ID, synthetic OOD, and RealNLR.

4.1 Dependence on Task Family

The gains depend on the task. Across the 129 answer-scored conditions at epoch 3, mean ID gains were +13.8 points for contour tracing (95% CI $[+9.0, +19.1]$), +13.5 for re-identification ($[+9.1, +18.2]$), and +17.0 for saccadic search ($[+10.7, +24.2]$). OOD gains were +13.0

([+8.6, +17.9]), +2.8 ([+1.6, +4.1]), and +7.0 ([+4.1, +10.0]). Re-identification loses most of its gain out of distribution. Contour tracing keeps it.

In saccadic search, strong answer-only models fail exactly where the requested depth passes the 2 to 4 hops seen in training (Figure 4, left). Process SFT shows that the route itself is not lost. InternVL3-14B writes the exact ID trajectory on 100% of puzzles and the exact OOD trajectory on 96.0%, but its full OOD output is correct only 30.0% of the time (Figure 4, right).

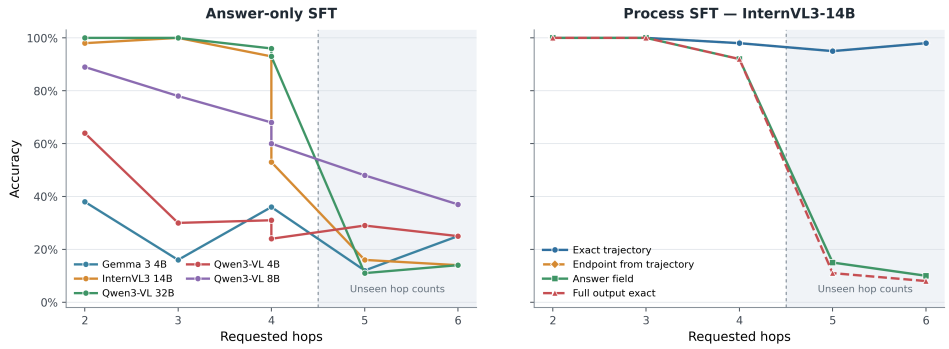


Figure 4: Saccadic search by requested hop count. Left: answer-only SFT. Right: InternVL3-14B process SFT, trajectory versus answer.

Of the 150 OOD examples, 99 have an exact trajectory and a wrong answer. In 87.9% of those 99, the answer names a color from earlier in the path. A wrong answer drawn uniformly from the seven incorrect colors would do so 42.0% of the time on average (per example, the share of incorrect colors that appear earlier in its trajectory), and 87.9% exceeds even the most conservative form of this null (binomial $p < 10^{-14}$). Because the answer is written before the trajectory (§3.3), it cannot read the generated path. The answer comes from a separate computation inside the forward pass, and that computation stops at a chain length seen in training, while the written steps continue to the end. Figure 17 (Appendix H) compares full-output and trajectory scoring for both process tasks.

4.2 REALNLR TRANSFER

Across 129 matched epoch-3 conditions (answer SFT, and process SFT scored on the answer), the mean change in RealNLR accuracy is -0.32 points (95% CI $[-1.50, +0.81]$). Synthetic ID gains correlate negatively with RealNLR gains for re-identification ($r = -0.34$) and saccadic search ($r = -0.30$), and positively for contour tracing ($r = +0.42$).

The zero mean hides many changes. On average an adapter fixes 28.4 of 300 RealNLR examples and breaks 29.3. We call $(\text{fixed} + \text{broken})/N$ the transfer turnover. Turnover rises with the synthetic ID gain ($r = 0.60$), and so do both parts: the rate of fixed base errors ($r = 0.60$) and the rate of

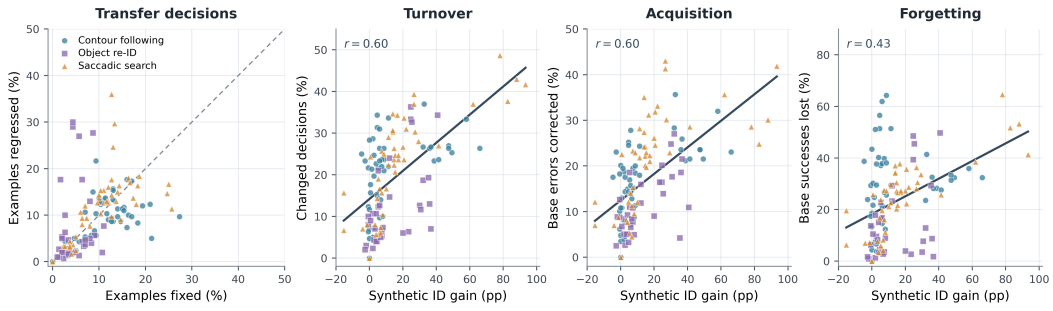


Figure 5: Transfer turnover. Left: RealNLR examples fixed versus broken, per condition. Right: synthetic ID gain against turnover, acquisition (base errors fixed), and forgetting (base successes lost).

broken base successes ($r = 0.43$) (Figure 5). The models that learn the synthetic task best change the most RealNLR answers, and they break nearly as many as they fix.

Language-side adaptation has a higher median than vision-side adaptation on synthetic ID for all three tasks (Figure 18, Appendix H). On RealNLR the difference is small. The median favors the language side by about 4 points for re-identification and is within 2 points of zero for the other two tasks.

4.3 Randomized Style and a Second Seed

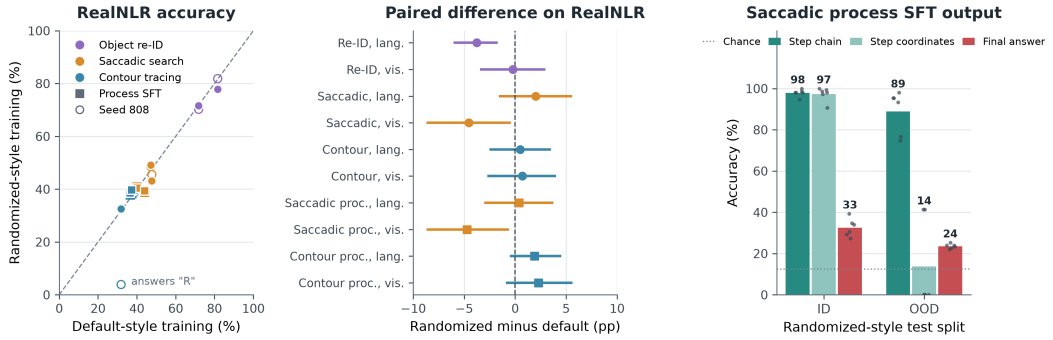


Figure 6: Randomized-style training, Qwen3-VL-8B. Left: RealNLR accuracy of each randomized run against its default-style control (seed 809, filled; randomized seed 808, open). Center: paired difference with 95% bootstrap CI over examples (lang. = language-side, vis. = matched vision-side). Right: saccadic process SFT, averaged over three regimes and two seeds.

Style randomization does not change transfer. For each of ten Qwen3-VL-8B conditions (five task and supervision pairs, two regimes), we compare the randomized run with the default-style run of the same seed on the same RealNLR examples. The mean paired difference is -0.6 points (95% t-interval over conditions $[-2.5, +1.4]$), and single conditions range from -4.7 to $+2.3$ (Figure 6, center). After an exact McNemar test for each condition and Benjamini–Hochberg correction for multiple comparisons, only one condition is significant: language-side re-identification, which loses 3.8 points with randomized training (27 examples fixed, 60 broken, of 877). The two styles still disagree on 19.8% of examples on average, which repeats the turnover pattern of §4.2: training changes which examples are right without changing how many. The randomized seed-808 runs fall on the same diagonal, with one outlier. The matched vision-side contour adapter drops to 3.9% on RealNLR, against 32.6% for the same condition at seed 809, because it answers "R", the label of the reference point, on 86.7% of items. In training its answers were component labels, so it learned an answer format that does not exist in the real prompt. Its synthetic accuracy is normal (59.3% ID).

Randomization did not stop Qwen3-VL-8B from learning the synthetic tasks. For answer SFT, randomized ID accuracy is within 10 points of the default-style control (median gap 2.7 points), and language-side re-identification models trained on randomized styles score 79 to 80% on default-style ID puzzles they never saw.

The readout gap appears in distribution. Randomized saccadic process SFT writes the correct color and shape chain for 98% of ID puzzles, with the correct grid coordinates for 97%, but the answer, written first, is correct for only 33% (Figure 6, right). On OOD the chain is still correct for 89% of puzzles, although the coordinates are right for only 14% (0% in four of six runs; the 5×5 grid is new). When the chain is correct and the answer wrong, the answer names an earlier color of the chain in 54.3% of OOD cases (332 of 611, pooled over six runs), against a 42.0% null ($p = 6.0 \times 10^{-10}$). This bias is weaker than the 87.9% of default-style InternVL3-14B (§4), but the gap between steps and answer is larger.

Single-seed runs vary. The seed-809 default-style runs repeat the seed-808 Qwen3-VL-8B conditions of the main matrix (Figure 13, Appendix F). Re-identification reproduces within 4 points on ID. But 5 of 15 conditions move by 10 points or more, including two of the largest seed-808 gains: language-side saccadic search falls from 78.7% to 39.3% ID, and its process SFT answer

from 98.0% to 56.7%, while its trajectory stays at 98.7%. Matched vision-side contour tracing rises from 14.0% to 60.0%. Randomized-style runs vary less across seeds (median 2.7 points, maximum 12). Any single condition in Figure 12 should be read with this spread in mind. The results we build on are aggregates over many conditions, or they hold at both seeds, namely the gap between steps and answer and the match between randomized and default-style RealNLR accuracy.

5 Effects of Fine-Tuning on Visual Reasoning

The analyses below use Qwen3-VL-8B at n=5,000, epoch 3, seed 808 (Appendix G), except where we name InternVL3-14B.

5.1 What object re-identification fine-tuning learned

The match relation is already in the base model. A linear classifier on pairwise features of the base model's hidden states for the two panels predicts whether the objects match with 99.1% accuracy on synthetic ID and 67.2% on synthetic OOD (Figure 7, left). Controlled identity and color changes confirm that the fine-tuned models rely on the object relation, not on simple cues.

Fine-tuning changes how this relation reaches the answer. We swap internal states between the base and fine-tuned models and measure the correct-answer log-probability margin. For language-side tuning, putting the fine-tuned final prompt state into the base model raises the margin by 3.3, and putting the base state into the fine-tuned model lowers it by 2.0; swapping visual states has little effect. Vision-side tuning shows the reverse. Visual-state transfer raises the margin by 2.2, and the reverse swap lowers it by 3.6. Language tuning changes a late decision step, and vision tuning changes the visual representation that step reads.

On real images the same transfers lower the margin, by 1.4 for the language side and 5.3 for the vision side (Figure 7, right). The mapping that fine-tuning fits to the synthetic distribution does not carry over to the relation in natural images.

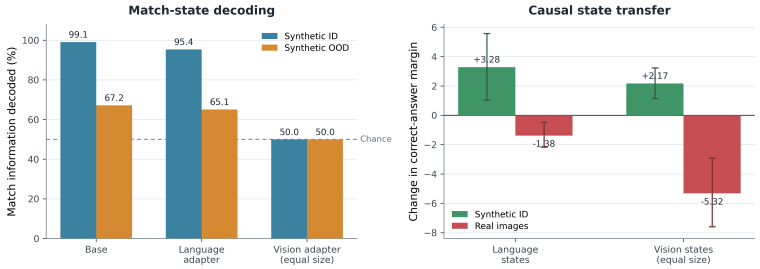


Figure 7: Object re-identification. Left: match information decoded from hidden states. Right: change in correct-answer margin from state transfer, on synthetic ID and real images.

5.2 What saccadic-search fine-tuning learned

Process supervision teaches the model to hop through the grid. Base InternVL3-14B gets the first transition right on 25.3% of ID puzzles and almost never writes a full trajectory (0.7%). After process SFT it writes nearly perfect trajectories on ID and OOD puzzles (Figure 8, left).

The learned transitions stay tied to the training queries. On an unchanged image, changing only the requested start drops a fine-tuned Qwen3-VL-8B from 75.0% to 52.1%, and changing only the requested hop count drops it to 31.2%.

On real images, one intervention locates the missing step. Naming and marking the correct first object more than doubles process-SFT accuracy on real two-hop puzzles, for example from 28.6% to 61.9% for the language side, while answer-only SFT does not change. Marking an unrelated object does not help, so the gain comes from the intermediate object, not from the marker. The randomized-style language-side process adapter repeats this on 42 real two-hop puzzles (26.2% to 52.4%, McNemar $p = 0.013$). Here the text naming the object carries the gain (57.1% with text

alone, 19.0% with the marker alone), the matched vision-side process adapter gains less (23.8% to 40.5%), and the base model gains as much as the language-side adapter (23.8% to 50.0%, $p = 0.035$). The step that fails on natural images is finding the next object and using it as the new start, and fine-tuning does not improve it.

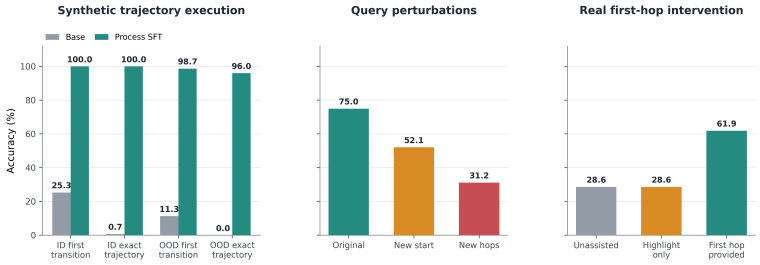


Figure 8: Saccadic search. Process SFT learns synthetic transitions (left) that depend on the training queries (center), but on real images it needs the first hop supplied (right).

5.3 What contour-tracing fine-tuning learned

Contour fine-tuning changes which image content controls the answer. In the base model, erasing the middle of the correct synthetic path has almost no effect on the answer margin, like erasing a matched distractor. After language-side answer SFT, and more strongly after process SFT, erasing the correct path lowers the margin while erasing the distractor does not (Figure 9). The matched vision-side answer adapter shows no change under either erasure; at seed 808 it scored 14.0% ID.

Path dependence does not show that the model traces a smooth contour, since landmarks along the path can also lead to the right endpoint. The real-image interventions separate these. A start segment, an endpoint, or isolated crossings give little reliable help. A long connected prefix of the correct contour, or the complete route, helps a lot, and a matched wrong route changes the answer.

The ability to use a supplied route is already in the base model. Drawing the complete route into the image helps base and fine-tuned models by similar amounts, and so does transferring the hidden state of a correctly marked route. Fine-tuning makes the answer depend more on the synthetic path, but finding the path in a natural image stays unreliable.

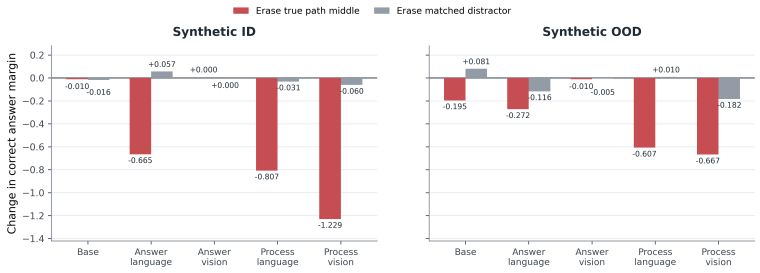


Figure 9: Change in the correct-answer margin when the middle of the true path or a matched distractor is erased. Fine-tuning makes contour answers depend on the true synthetic path.

5.4 Three stages of nonlocal reasoning

Each task needs the model to construct a visual state, operate on it, and read out an answer. In re-identification, the model must represent both objects, compare them under a transformation, and answer yes or no. In saccadic search, it must find the current cell, read its label, and move on. In contour tracing, it must pick out the path and follow it through crossings. In every task the base model already has part of the procedure: the match relation, access to the start object, or the use of a supplied route (Figure 10). Fine-tuning improves the next stage on synthetic images: the yes/no readout, the transition between cells, or dependence on the true path. What still fails on RealNLR is

the first stage, constructing the state from a natural image, and neither fine-tuning nor randomized rendering (§4.3) improves it.

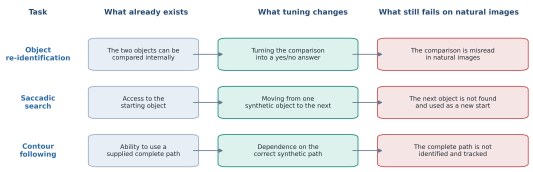


Figure 10: What the base model already has, what fine-tuning changes, and what still fails on natural images.

6 Limitations and Future Work

Each RealNLR family comes from a single source dataset, and two of them are not photographs. Contour tracing composites simulated wires onto real backgrounds, and re-identification uses synthetic scenes rendered for photorealism. The main matrix uses one seed. The second seed shows that single conditions can move by up to 46 points, so per-condition numbers are less reliable than aggregates. The style study covers one model (Qwen3-VL-8B), and it has no base-model evaluation on the full RealNLR sets, so it compares training styles with each other rather than with the base model. Its saccadic process target adds grid coordinates, so that condition is not identical to its control. Process targets write the answer before the steps. A steps-first target, which lets the answer condition on the generated trajectory, may close the gap between steps and answer, and we did not test it. We also did not test reinforcement learning. Given evidence that SFT memorizes where reinforcement learning generalizes (Chu et al., 2025), it is the next training signal to try.

7 Conclusion

We ask whether nonlocal visual reasoning tasks (saccadic search, contour tracing, and object re-identification) can be taught by fine-tuning, and find what the model actually learns. We fine-tune open VLMs on synthetic versions of these tasks, varying the adapted side, the supervision format, and the data scale, and we introduce RealNLR, a benchmark that poses the same tasks on realistic imagery. Fine-tuning solves the synthetic tasks, but the gains halve on hard instances and net to zero on realistic images, where adapters fix as many examples as they break. Process supervision teaches the transitions of a search well enough to reproduce paths longer than any seen in training, yet the model still reports the answer at the chain length it was trained on. Training on renderings with randomized backgrounds, colors, fonts, and layouts leaves these results unchanged. Our internal analyses show why: these tasks require constructing the relevant state from the image, operating on that state, and converting the result into an answer, and fine-tuning improves only one of these stages. Our results point to VLMs learning the dataset rather than the algorithm: their visual reasoning remains a property of their training data, not a skill they carry into new images.

AI USE STATEMENT

We used generative AI tools only to edit this paper for readability and for parts of the LaTeX formatting. We did not use them to generate datasets (all synthetic data come from procedural generators, and scoring is deterministic with no LLM judge), to develop the conceptual framework, to propose or refine hypotheses, to design experiments, to implement methods, to clean or reformat data, or to interpret results. Formulating or proving mathematical claims, translation, and qualitative data analysis are not applicable to this work. We reviewed all AI-assisted work and take responsibility for the final content of this paper, including text edited with the aid of generative AI.

ETHICS STATEMENT

This work fine-tunes open models on synthetic puzzles and evaluates them on RealNLR, which we derive from publicly released, annotated datasets used under their licenses (Appendices B and C). The object re-identification source scenes are synthetic renderings, so RealNLR adds no images of real persons. The human accuracy figures come from the authors; we recruited no external participants. We release no high-risk models and no scraped data.

REPRODUCIBILITY STATEMENT

Section 3 gives the data splits, the fine-tuning regimes, and the evaluation procedure. Appendix A lists all prompts, Appendix B gives the RealNLR construction parameters, Appendix C gives the training hyperparameters, seeds, hardware, and statistical procedures, Appendix D gives the trainable parameter counts, Appendix G gives the layers and regions used in each mechanistic analysis, and Appendix F gives every randomized-style and seed-809 result. The RealNLR benchmark is at the anonymous link in the abstract. The data release records the generator settings and the realized difficulty of each example. We will release the per-example evaluation outputs and the analysis script.

REFERENCES

Shuai Bai et al. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Shmuel Berman and Jia Deng. VLMs have tunnel vision: Evaluating nonlocal visual reasoning in leading VLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://arxiv.org/abs/2507.13361.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs, 2023. URL https://arxiv.org/abs/2305.14314.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://arxiv.org/abs/2004.07780.

Gemma Team et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Hyesoo Hong, Minsoo Kim, Wonje Jeung, Sangyeon Yoon, Dongjae Jeon, and Albert No. VLMs trace without tracking: Diagnosing failures in visual path following, 2026. URL https://arxiv.org/abs/2605.15672.

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.

Kung-Hsiang Huang, Can Qin, Haoyi Qiu, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. Why vision language models struggle with visual arithmetic? Towards enhanced chart and geometry understanding, 2025. URL https://arxiv.org/abs/2502.11492.

Ryo Kamoi, Yusen Zhang, Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, and Rui Zhang. VisOnlyQA: Large vision language models still struggle with visual perception of geometric information. In *Conference on Language Modeling (COLM)*, 2025. URL https://arxiv.org/abs/2412.00947.

Min Je Kim, Muhammad Munsif, Altaf Hussain, Hikmat Yar, and Sung Wook Baik. Pseudo-labeling driven refinement of benchmark object detection datasets via analysis of learning patterns, 2025. URL https://arxiv.org/abs/2506.00997.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words, 2024. URL https://arxiv.org/abs/2407.06581.

Massimo Rizzoli, Simone Alghisi, Seyed Mahed Mousavi, and Giuseppe Riccardi. Synthetic stimuli, real gains: Rethinking VLM fine-tuning through fully controlled data generation, 2025. URL https://arxiv.org/abs/2511.11440.

Luca M. Schulze Buschoff, Konstantinos Voudouris, Elif Akata, Matthias Bethge, Joshua B. Tenenbaum, and Eric Schulz. Testing the limits of fine-tuning for improving visual cognition in vision language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.15678.

Haz Sameen Shahgir, Xiaofu Chen, Yu Fu, Erfan Shayegani, Nael Abu-Ghazaleh, Yova Kementchedjhieva, and Yue Dong. VLMs need words: Vision language models ignore visual detail in favor of semantic anchors. In *Conference on Language Modeling (COLM)*, 2026. URL https://arxiv.org/abs/2604.02486.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2403.16999.

Zheng Tang et al. The 9th AI city challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 5526–5535, 2025.

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. 6209–6218, 2024. URL https://arxiv.org/abs/2401.06209.

Shuo Wang et al. The 8th AI city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 7261–7272, 2024a.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Yongyuan Liang, Yuhang Zhou, Xiaoyu Liu, Ziyi Zang, Ming Li, Chung-Ching Lin, Kevin Lin, Linjie Li, Furong Huang, and Lijuan Wang. ViCrit: A verifiable reinforcement learning proxy task for visual perception in VLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://arxiv.org/abs/2506.10128.

Yizhou Wang, Tim Meinhardt, Orcun Cetintas, Cheng-Yen Yang, Sameer Satish Pusegaonkar, Benjamin Missaoui, Sujit Biswas, Zheng Tang, and Laura Leal-Taixé. MCBLT: Multi-camera multi-object 3D tracking in long videos, 2024b. URL https://arxiv.org/abs/2412.00692.

Yi Yang, Xiang Fei, Lehong Wang, Chenhao Li, Zilin Dai, Henry Kou, Lu Li, and Howie Choset. DeformX: A versatile co-simulation framework for deformable linear objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2026. URL https://arxiv.org/abs/2606.22116.

Guanyu Zhou, Yida Yin, Wenhao Chai, Shengbang Tong, Xingyu Fu, and Zhuang Liu. VisionFoundry: Teaching VLMs visual perception with synthetic images, 2026. URL https://arxiv.org/abs/2604.09531.

Jinguo Zhu et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

A Prompts Used

A.1 Synthetic answer-only prompts

Object re-identification.

> *The image has two panels. Decide whether the object in IMAGE 2 is the same object as IMAGE 1 after a possible whole-object rotation, scale, or translation. Ignore extra distractor objects. Answer only yes or no.*

Saccadic search.

> *Each cell contains a colored shape and a label naming the next colored shape. Start at {start color} {start shape}. Follow labels for {number of steps} steps. What color is the final shape? Answer only the full color name.*

Contour following.

> *Which component does the wire from port {port number} on the breadboard, which is the gray rectangle with numbered ports, connect to? A wire is a series of connected, same-colored line segments that go from the center of one white circular port to another port. Each wire only connects two ports, one at either end. Wires do not change colors. Answer only the component label.*

A.2 Synthetic process-supervision prompts

Saccadic search with process supervision.

> *Each cell contains a colored shape and a label naming the next colored shape. Start at {start color} {start shape}. Follow labels for {number of steps} steps. What color is the final shape? Return JSON with trajectory (ordered shape/color objects) and answer.*

Contour following with process supervision.

> *Which component does the wire from port {port number} on the breadboard, which is the gray rectangle with numbered ports, connect to? All wires are black. A wire is a connected series of line segments from one white circular port to another port. Crossing wires do not connect. Each wire has only two end ports. Follow the queried wire from the breadboard port to the component. Return JSON with the marker labels on that wire in order from the queried port to the component and the answer, using fields markers and answer.*

In both process targets the JSON keys are sorted, so the target (and every fine-tuned model output we checked) has the form `{"answer": ..., "trajectory": [...]}` or `{"answer": ..., "markers": [...]}`.

A.3 REAL-IMAGE TRANSFER PROMPTS

Object re-identification.

> *The first image is a close-up reference crop of one object. The second image is a later scene containing several boxed, lettered candidates. Its appearance may have shifted (angle, lighting, partial occlusion). Exactly one lettered box is the same physical object as the reference. Which letter is it? Answer with a single letter.*

Saccadic search. Prompts varied with start object and total number of hops. One complete example was:

> *Begin at the car. Follow the label to the next object, and keep following labels. Then read that object's label and hop once more. After 2 hops total, what object are you on? Please provide the answer only as the object name.*

Contour following.

> *Point R is marked on a cable, wire, rope, or similar elongated object. Several other points are marked with letters. Starting from R, follow the same physical object through the image. At any crossing, overlap, or contact with another object, continue along the object that R is on — do not switch to a different one. Exactly one lettered point is marked on that same object. Which letter is it? Answer with a single letter only.*

B REALNLR DETAILS

B.1 DATASET CONSTRUCTION

Saccadic search. Images and annotations come from MJ-COCO-2025 (Kim et al., 2025). We remove duplicate objects, filter all objects by size, and sample a sparse subset per image (Table 1). Each object is labeled with the next object in a hop chain, never its own name. We check that every object has one outgoing label, that no object labels itself, and that the stated hop count from the start reaches the answer. Images are cropped around the sampled objects and upscaled.

Contour tracing. We use the synthetic partition of WireSeg-36K (Yang et al., 2026) (simulated wires composited on real backgrounds). Crossings and candidate points are sampled by proximity, and distractors match the target wire in color and distance (Table 1). We drop any set where the correct answer is the nearest or farthest option, so distance heuristics fail. Images are cropped and upscaled.

Object re-identification. We use NVIDIA's PhysicalAI-SmartSpaces (MTMC_Tracking_ 2026) (Tang et al., 2025; Wang et al., 2024a;b), a synthetic partition rendered through Cosmos Transfer for photorealism. Identity comes from the dataset's annotations. Reference and query frames are sampled per track (Table 1), and the query scene shows candidates of the same category below a fixed overlap threshold. Cropping the query to the candidate region and upscaling it was the largest legibility fix.

Source licenses. MJ-COCO-2025 is released under the MIT license (its images inherit the MS-COCO terms), and WireSeg-36K and PhysicalAI-SmartSpaces under CC-BY-4.0.

B.2 REALNLR DATASET INFORMATION

B.3 HUMAN VERIFICATION AND ACCURACY

We measured human accuracy (Figure 11) on 30 samples each for saccadic search (24/30, 80.0%; 95% Wilson CI 62.7–90.5%) and contour tracing (27/30, 90.0%; 95% Wilson CI 74.4–96.5%). Ob-

Table 1: RealNLR construction parameters per task family.

Parameter	Saccadic Search	Contour Tracing	Object Re-ID
Delivered size	549	1000	877
Candidate/object count (min)	3	2	2
Candidate/object count (max)	4	5	4
Chance baseline (%)	32.0	31.9	29.8
Min box/object size (px)	70	N/A	40
Hop count / frame gap (min)	2	5	10
Hop count / frame gap (max)	3	N/A	300
Color match tolerance (RGB Δ)	N/A	40	N/A
Crossing distance threshold (px)	N/A	12	N/A
Line proximity threshold (px)	N/A	15	N/A
Marker separation threshold (px)	N/A	25	N/A
Max IoU overlap	N/A	N/A	0.1
Crop upscale target, primary (px)	700	500	800
Crop upscale target, secondary (px)	N/A	N/A	350
Crop margin	25% + 60px	N/A	N/A
Guess-resistance enforcement	Universal labeling	Hard rank-rejection (closest+farthest)	Same-category + IoU cap

ject re-identification used a different procedure: we audited all 877 puzzles by hand during construction and corrected or removed any a human could not solve. The 100% figure for that family is a verification pass that guarantees solvability, not an independent measurement of human accuracy.

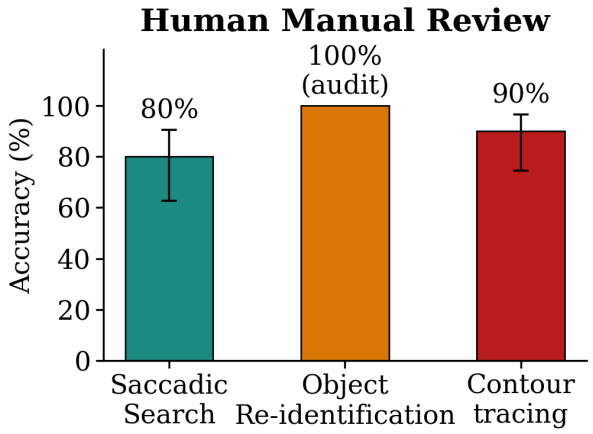


Figure 11: Human performance on RealNLR. Saccadic search and contour tracing show measured accuracy on 30 samples each, with 95% Wilson intervals. The object re-identification bar is the construction-time audit of all 877 examples, not an independent measurement.

C Implementation Details

Models were fine-tuned for three epochs with paged 8-bit AdamW, learning rate 1×10^{-4}, cosine schedule, warmup ratio 0.03, and no weight decay. The per-device batch size was 1 with 16 gradient accumulation steps (effective batch size 16). Models were loaded in 4-bit NF4 with double quantization (Dettmers et al., 2023) and trained in bfloat16 where supported and float16 otherwise.

LoRA (Hu et al., 2021) adapters used rank $r = 16$, $\alpha = 32$, and dropout 0 unless stated otherwise. For language-side adaptation, LoRA was applied to the language model's attention and MLP projections. The base weights were frozen in every regime, and the multimodal projector was frozen in every regime except connector-only, which applies LoRA ($r = 16$) to the two linear layers of the vision-to-language merger (for Qwen3-VL-8B, the main merger and the three deep-stack merg-

ers; 1.15M trainable parameters). For the parameter-matched Qwen3-VL 8B vision-side condition, we raised the rank to $r = 79$ with $\alpha = 158$, which matches the trainable parameter count of the language-side condition to within 0.1%. The main matrix used training and data seed 808; the style study and the replication in §4.3 add seed 809. Randomized styles were drawn with a fixed style seed. Training ran on NVIDIA A100, H100, L4, T4, and RTX 4090 GPUs.

All assets were used under their licenses: the Tunnel Vision generators (Berman & Deng, 2025) are BSD-3-Clause, the Qwen3-VL family and InternVL3 14B are Apache-2.0, and Gemma 3 4B IT follows the Gemma Terms of Use.

For mechanistic interventions, we score each candidate answer by summing its token log-probabilities under teacher forcing, and define the correct-answer margin as the correct candidate's score minus the highest-scoring incorrect candidate.

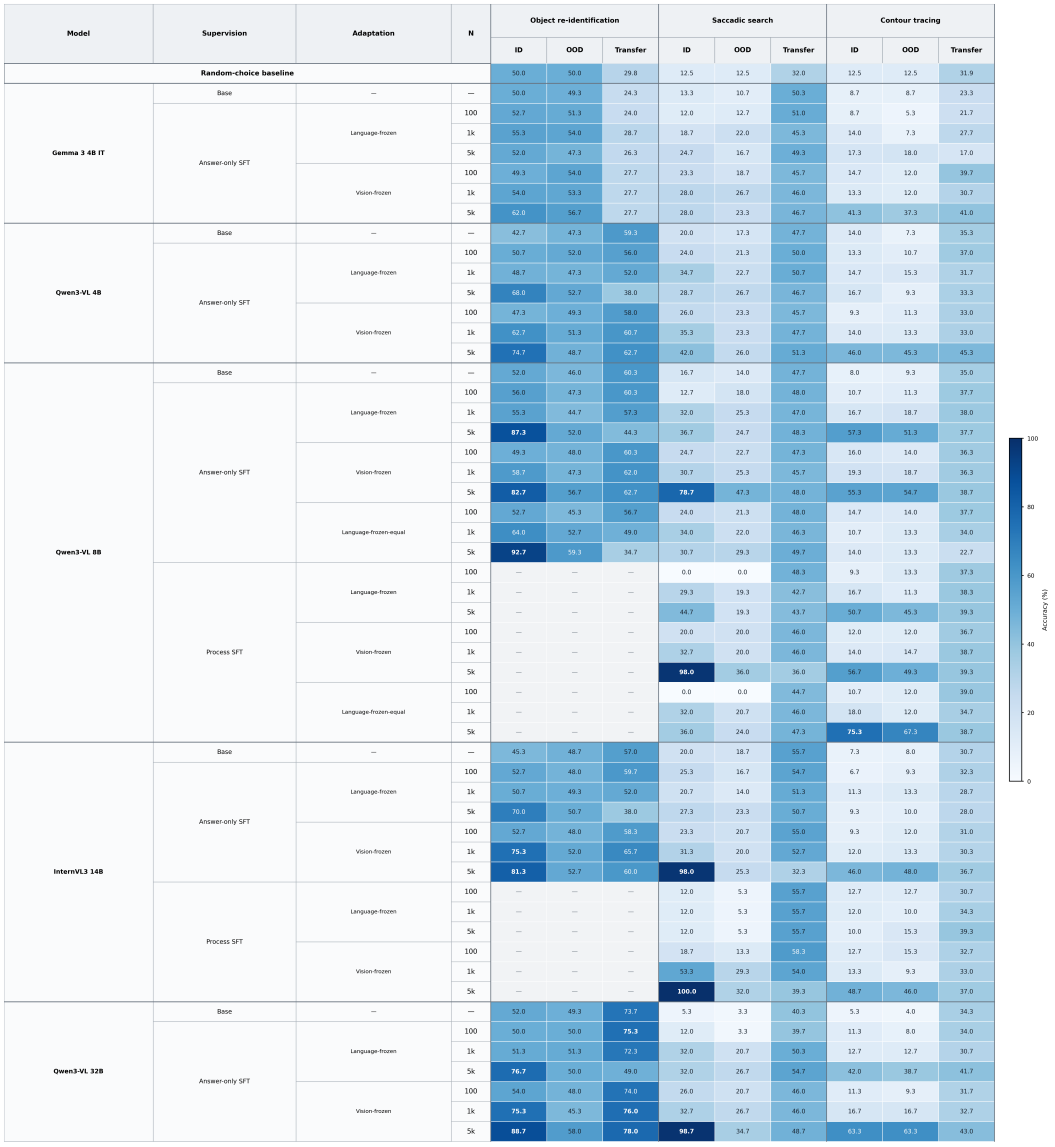
Statistical reporting. Unless stated otherwise, intervals are two-sided 95% percentile-bootstrap confidence intervals with 2,000 resamples. For aggregate fine-tuning gains we resample matched conditions; for mechanistic interventions and the per-condition RealNLR differences of §4.3, we resample examples. The mean over the ten style-study conditions uses a t-interval. Per-condition RealNLR differences in §4.3 use exact McNemar tests with Benjamini–Hochberg correction over the ten conditions. Mechanistic p-values use paired random-sign permutation tests with 2,000 permutations and Benjamini–Hochberg correction. Box plots show the median and interquartile range, with whiskers at $1.5\times$ the interquartile range. Intervals quantify variation across conditions or examples, not across training seeds; §4.3 reports seed variation separately. Pearson correlations are descriptive.

D Parameter Counts of Models Fine-tuned (Regime + SFT type)

Table 2: Total and trainable parameter counts per model, SFT type, and regime. "Language frozen" is vision-side adaptation and "vision frozen" is language-side adaptation. Connector-only runs train only the merger (§3.3).

Model	SFT type	Regime	Total model params	Trainable params
Gemma 3 4B IT	Answer-only SFT	Language frozen	2.499B	8.70M
Gemma 3 4B IT	Answer-only SFT	Vision frozen	2.520B	29.80M
Qwen3-VL 4B Instruct	Answer-only SFT	Language frozen	2.423B	7.24M
Qwen3-VL 4B Instruct	Answer-only SFT	Vision frozen	2.449B	33.03M
Qwen3-VL 8B Instruct	Answer-only SFT	Language frozen	5.017B	8.85M
Qwen3-VL 8B Instruct	Answer-only SFT	Vision frozen	5.052B	43.65M
Qwen3-VL 8B Instruct	Answer-only SFT	Language frozen equal	5.052B	43.68M
Qwen3-VL 8B Instruct	Process SFT	Language frozen	5.017B	8.85M
Qwen3-VL 8B Instruct	Process SFT	Vision frozen	5.052B	43.65M
Qwen3-VL 8B Instruct	Process SFT	Language frozen equal	5.052B	43.68M
Qwen3-VL 8B Instruct	Answer / process SFT	Connector only	–	1.15M
InternVL3 14B	Answer-only SFT	Language frozen	8.343B	6.29M
InternVL3 14B	Answer-only SFT	Vision frozen	8.405B	68.81M
InternVL3 14B	Process SFT	Language frozen	8.343B	6.29M
InternVL3 14B	Process SFT	Vision frozen	8.405B	68.81M
Qwen3-VL 32B Instruct	Answer-only SFT	Language frozen	17.468B	8.91M
Qwen3-VL 32B Instruct	Answer-only SFT	Vision frozen	17.594B	134.22M

E HEATMAP OF EPOCH 3 ACCURACIES



Model	Supervision	Adaptation	N	Object re-identification			Saccadic search			Contour tracing		
				ID	OOD	Transfer	ID	OOD	Transfer	ID	OOD	Transfer
Random-choice baseline				50.0	50.0	29.8	12.5	12.5	32.0	12.5	12.5	31.9
Gemma 3 4B IT	Base	—	—	50.0	49.3	24.3	13.3	10.7	50.3	8.7	8.7	23.3
	Answer-only SFT	Language-frozen	100	52.7	51.3	24.0	12.0	12.7	51.0	8.7	5.3	21.7
			1k	55.3	54.0	28.7	18.7	22.0	45.3	14.0	7.3	27.7
			5k	52.0	47.3	26.3	24.7	16.7	49.3	17.3	18.0	17.0
		Vision-frozen	100	49.3	54.0	27.7	23.3	18.7	45.7	14.7	12.0	39.7
			1k	54.0	53.3	27.7	28.0	26.7	46.0	13.3	12.0	30.7
			5k	62.0	56.7	27.7	28.0	23.3	46.7	41.3	37.3	41.0
Qwen3-VL 4B	Base	—	—	42.7	47.3	59.3	20.0	17.3	47.7	14.0	7.3	35.3
	Answer-only SFT	Language-frozen	100	50.7	52.0	56.0	24.0	21.3	50.0	13.3	10.7	37.0
			1k	48.7	47.3	52.0	34.7	22.7	50.7	14.7	15.3	31.7
			5k	68.0	52.7	38.0	28.7	26.7	46.7	16.7	9.3	33.3
		Vision-frozen	100	47.3	49.3	58.0	26.0	23.3	45.7	9.3	11.3	33.0
			1k	62.7	51.3	60.7	35.3	23.3	47.7	14.0	13.3	33.0
			5k	74.7	48.7	62.7	42.0	26.0	51.3	46.0	45.3	45.3
Qwen3-VL 8B	Base	—	—	52.0	46.0	60.3	16.7	14.0	47.7	8.0	9.3	35.0
	Answer-only SFT	Language-frozen	100	56.0	47.3	60.3	12.7	18.0	48.0	10.7	11.3	37.7
			1k	55.3	44.7	57.3	32.0	25.3	47.0	16.7	18.7	38.0
			5k	**87.3**	52.0	44.3	36.7	24.7	48.3	57.3	51.3	37.7
		Vision-frozen	100	49.3	48.0	60.3	24.7	22.7	47.3	16.0	14.0	36.3
			1k	58.7	47.3	62.0	30.7	25.3	45.7	19.3	18.7	36.3
			5k	**82.7**	56.7	62.7	**78.7**	47.3	48.0	55.3	54.7	38.7
		Language-frozen-equal	100	52.7	45.3	56.7	24.0	21.3	48.0	14.7	14.0	37.7
			1k	64.0	52.7	49.0	34.0	22.0	46.3	10.7	13.3	34.0
			5k	**92.7**	59.3	34.7	30.7	29.3	49.7	14.0	13.3	22.7
	Process SFT	Language-frozen	100	—	—	—	0.0	0.0	48.3	9.3	13.3	37.3
			1k	—	—	—	29.3	19.3	42.7	16.7	11.3	38.3
			5k	—	—	—	44.7	19.3	43.7	50.7	45.3	39.3
		Vision-frozen	100	—	—	—	20.0	20.0	46.0	12.0	12.0	36.7
			1k	—	—	—	32.7	20.0	46.0	14.0	14.7	38.7
			5k	—	—	—	**98.0**	36.0	36.0	56.7	49.3	39.3
		Language-frozen-equal	100	—	—	—	0.0	0.0	44.7	10.7	12.0	39.0
			1k	—	—	—	32.0	20.7	46.0	18.0	12.0	34.7
			5k	—	—	—	36.0	24.0	47.3	**75.3**	67.3	38.7
InternVL3 14B	Base	—	—	45.3	48.7	57.0	20.0	18.7	55.7	7.3	8.0	30.7
	Answer-only SFT	Language-frozen	100	52.7	48.0	59.7	25.3	16.7	54.7	6.7	9.3	32.3
			1k	50.7	49.3	52.0	20.7	14.0	51.3	11.3	13.3	28.7
			5k	70.0	50.7	38.0	27.3	23.3	50.7	9.3	10.0	28.0
		Vision-frozen	100	52.7	48.0	58.3	23.3	20.7	55.0	9.3	12.0	31.0
			1k	**75.3**	52.0	65.7	31.3	20.0	52.7	12.0	13.3	30.3
			5k	**81.3**	52.7	60.0	**98.0**	25.3	32.3	46.0	48.0	36.7
	Process SFT	Language-frozen	100	—	—	—	12.0	5.3	55.7	12.7	12.7	30.7
			1k	—	—	—	12.0	5.3	55.7	12.0	10.0	34.3
			5k	—	—	—	12.0	5.3	55.7	10.0	15.3	39.3
		Vision-frozen	100	—	—	—	18.7	13.3	58.3	12.7	15.3	32.7
			1k	—	—	—	53.3	29.3	54.0	13.3	9.3	33.0
			5k	—	—	—	**100.0**	32.0	39.3	48.7	46.0	37.0
Qwen3-VL 32B	Base	—	—	52.0	49.3	73.7	5.3	3.3	40.3	5.3	4.0	34.3
	Answer-only SFT	Language-frozen	100	50.0	50.0	**75.3**	12.0	3.3	39.7	11.3	8.0	34.0
			1k	51.3	51.3	72.3	32.0	20.7	50.3	12.7	12.7	30.7
			5k	**76.7**	50.0	49.0	32.0	26.7	54.7	42.0	38.7	41.7
		Vision-frozen	100	54.0	48.0	74.0	26.0	20.7	46.0	11.3	9.3	31.7
			1k	**75.3**	45.3	**76.0**	32.7	26.7	46.0	16.7	16.7	32.7
			5k	**88.7**	58.0	**78.0**	**98.7**	34.7	48.7	63.3	63.3	43.0

Figure 12: All epoch-3 accuracies of the main matrix (default style, seed 808, 300 RealNLR questions per task). For process-SFT rows, the synthetic ID and OOD cells give the answer accuracy extracted from the process output; transfer cells give answer accuracy on RealNLR. Dashes mark combinations that were not run.

F RANDOMIZED-STYLE AND SEED-809 RESULTS

Table 3 lists every Qwen3-VL-8B run of §4.3. Randomized runs are tested on the randomized splits and default-style runs on the default splits. RealNLR columns use the full sets. The default-style regime without a randomized counterpart (unmatched vision-side) and the randomized connector-only regime without a default-style counterpart are listed but not used in the paired comparison.

Cross-style transfer. Three randomized re-identification adapters were also tested on the default-style splits: language-side seed 808 scores 80.0% ID and 58.0% OOD (88.7% and 52.7% in random-

Table 3: Qwen3-VL-8B, n=5,000, epoch 3. Accuracy (%) on synthetic ID and OOD (in the training style) and on the full RealNLR set. Process rows give answer accuracy.

		Randomized, seed 808			Randomized, seed 809			Default, seed 809		
Task	Regime	ID	OOD	RealNLR	ID	OOD	RealNLR	ID	OOD	RealNLR
Object re-ID	Language-side	88.7	52.7	81.9	84.7	52.0	77.9	81.3	53.3	81.6
Object re-ID	Vision-side (matched)	90.0	57.3	70.2	90.7	58.0	71.6	88.7	57.3	71.8
Object re-ID	Vision-side	–	–	–	–	–	–	86.0	54.0	78.9
Object re-ID	Connector only	76.7	52.0	77.7	75.3	50.0	79.7	–	–	–
Saccadic search	Language-side	37.3	24.7	48.6	49.3	32.7	49.2	39.3	26.7	47.2
Saccadic search	Vision-side (matched)	32.0	27.3	45.5	26.7	22.0	43.2	36.7	26.7	47.7
Saccadic search	Vision-side	–	–	–	–	–	–	31.3	25.3	45.9
Saccadic search	Connector only	32.0	26.0	47.2	26.0	24.7	46.3	–	–	–
Contour tracing	Language-side	51.3	50.7	38.8	49.3	46.7	39.4	51.3	56.0	38.9
Contour tracing	Vision-side (matched)	59.3	49.3	3.9	60.7	52.7	32.6	60.0	54.7	31.9
Contour tracing	Vision-side	–	–	–	–	–	–	44.0	50.0	38.0
Contour tracing	Connector only	37.3	28.0	35.2	35.3	28.7	33.5	–	–	–
Saccadic search (process)	Language-side	34.0	23.3	40.8	39.3	22.0	40.4	56.7	26.7	40.1
Saccadic search (process)	Vision-side (matched)	34.7	22.7	38.8	30.7	24.0	39.3	40.7	30.0	44.1
Saccadic search (process)	Vision-side	–	–	–	–	–	–	34.7	27.3	43.4
Saccadic search (process)	Connector only	27.3	24.0	44.6	29.3	25.3	46.6	–	–	–
Contour tracing (process)	Language-side	61.3	47.3	37.8	59.3	45.3	38.5	50.7	42.7	36.6
Contour tracing (process)	Vision-side (matched)	69.3	54.0	38.2	58.7	42.0	39.7	68.7	54.7	37.4
Contour tracing (process)	Vision-side	–	–	–	–	–	–	43.3	34.0	36.1
Contour tracing (process)	Connector only	18.7	11.3	34.2	21.3	12.7	33.7	–	–	–

ized style), language-side seed 809 scores 79.3% and 53.3%, and connector-only seed 808 scores 70.0% and 51.3%.

Mechanistic replication. We repeated the Appendix G diagnostics on the randomized seed-808 adapters with the same layers and regions. Besides the saccadic first-hop result in §5, the contour erasure pattern repeats. On 48 synthetic ID examples, erasing the middle of the true path changes the correct-answer margin by $+0.09$ in the base model and by -0.87 to -0.92 in all four fine-tuned adapters, while erasing a matched distractor changes it by less than 0.07 in every model. Here the matched vision-side answer adapter, which unlike its default-style seed-808 counterpart learned the synthetic task (59.3% ID), shows the same dependence.

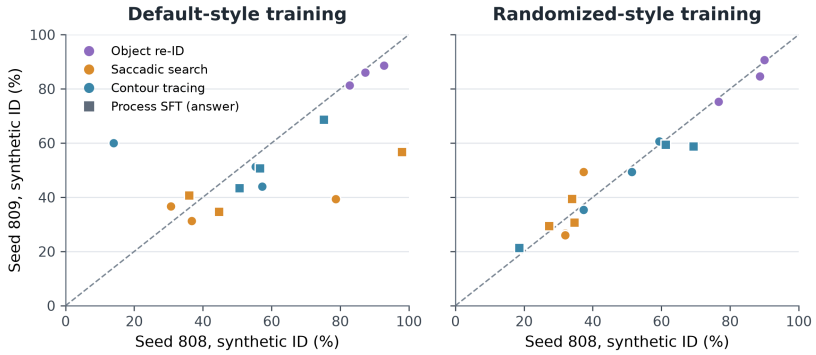


Figure 13: Synthetic ID accuracy of the same Qwen3-VL-8B condition at seeds 808 and 809. Left: default-style training (seed 808 values are those of Figure 12). Right: randomized-style training.

G Mechanistic Analysis Details

All mechanistic analyses use Qwen3-VL-8B-Instruct with 5,000 training examples, epoch 3, and seed 808. The base model is evaluated with adapters disabled. Hidden states come from the language transformer, and visual regions are mapped to visual-token positions with the processor's image-token mask.

Linear probe. For object re-identification, we probe whether the match relation can be decoded from the base model's hidden states at language layer 24. For each panel pair, the feature concatenates $|h_1 - h_2|$ and $h_1 \odot h_2$, where h_1 and h_2 are the panel representations. We fit a `StandardScaler` followed by logistic regression ($C = 1.0$, `class_weight=balanced`, `max_iter=4000`) on a synthetic-ID discovery split. Layer 24 was selected from $\{12, 18, 24, 30, 35\}$ on a separate synthetic-ID confirmation split; OOD and real examples were not used for layer selection. The confirmation split has 108 synthetic-ID pairs and the OOD evaluation 192 pairs.

Activation transfer. A receiver example is first evaluated normally. Then the selected receiver activation is replaced with the corresponding source activation, at the final prompt position, at all visual tokens, or at the visual tokens of a selected task region. Shuffling the source visual-token order is a positional control. State transfer replaces activations; it does not add a steering vector.

Object re-identification. Synthetic scenes are paired into controlled positive and negative identity swaps and color controls, with 12 confirmation pairs per counterfactual family. Language-side and vision-side tuning are analyzed separately. The selected layer and region are layer 18 with second-panel visual tokens for the base model, layer 12 for language-side answer SFT, and layer 18 for parameter-matched vision-side answer SFT. A separate steering experiment adds the normalized probe direction to the final-prompt hidden state at layers 2, 20, and 24 and compares it with a random vector of the same norm.

Saccadic search. Counterfactual pairs come from the same synthetic scene with a changed start object, hop count, or trajectory prefix, and the same scene graph. Twelve confirmation pairs are used per family. Source activations are transferred at the final-prompt hidden state, all visual tokens, or the visual tokens of a selected region. The selected layers, all using the visual tokens of the endpoint cell, are 0 (base), 24 (language-side answer SFT), 18 (matched vision-side answer SFT), 12 (language-side process SFT), and 35 (vision-side process SFT).

Contour following. Synthetic counterfactuals change the queried path to a length- and crossing-matched distractor or a harder path, with 12 confirmation pairs per family. Pixel interventions erase the middle of the correct path, erase a matched distractor, keep only endpoints, or highlight the target route. On realistic images, interventions use endpoint or annotation-based highlighting when a full ground-truth path is not available. The selected layers, all using the visual tokens of the target path, are 6 (base), 6 (language-side answer SFT), 12 (matched vision-side answer SFT), 6 (language-side process SFT), and 18 (vision-side process SFT). A separate restoration analysis copies visual-token activations from a correctly marked route into the receiver positions and compares them with activations from matched wrong routes.

H ADDITIONAL FIGURES

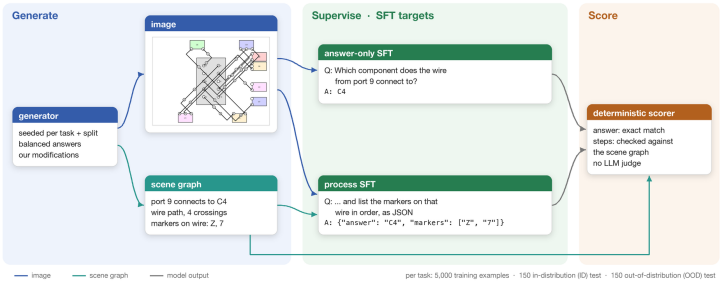


Figure 14: From generator to score. Both SFT targets come from the same image, and the scorer checks the model's steps against the scene graph.

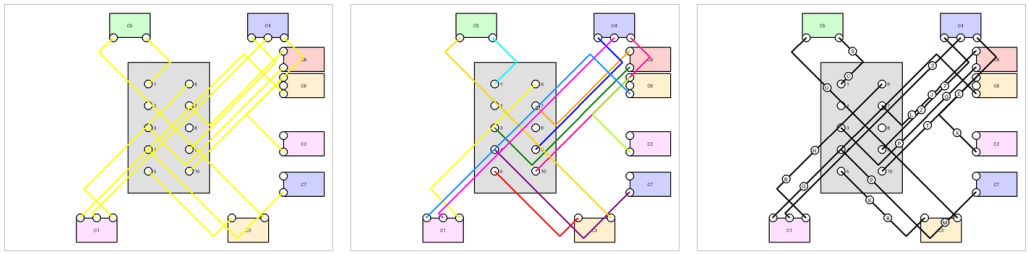


Figure 15: The same circuit scene in its three renderings. A model that improves only when every wire has its own color has learned color matching, not tracing. The black-wire version adds the markers that process-SFT targets list in order.

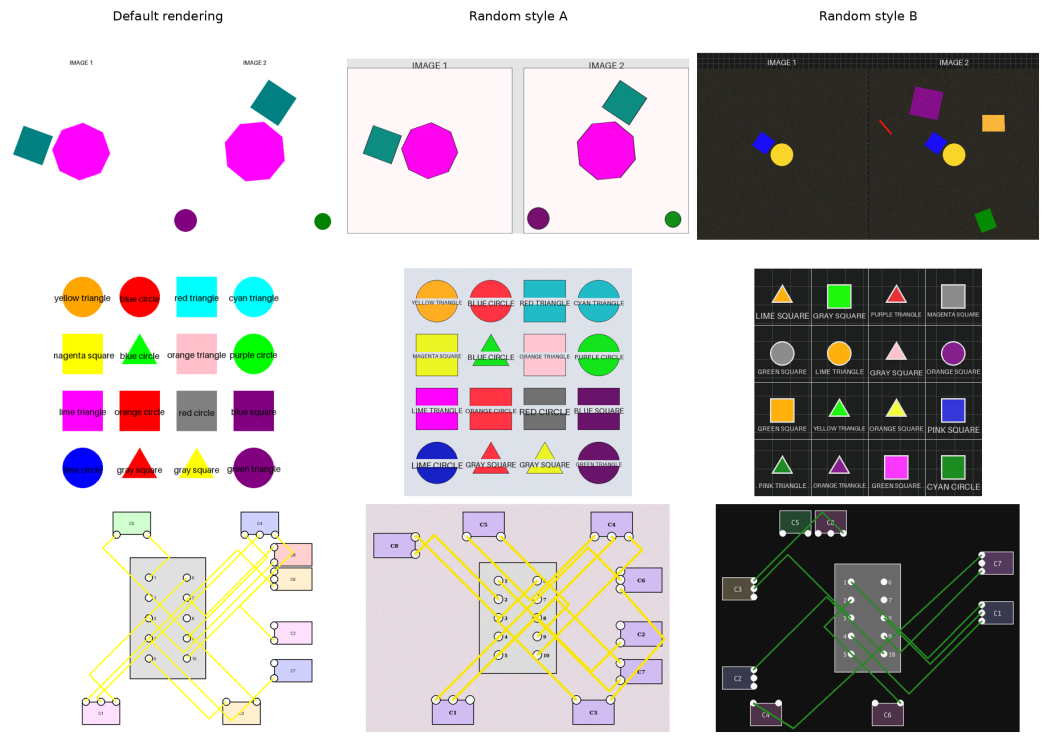


Figure 16: Randomized style. Each row is one task. Left: the default rendering of one held-out item. Middle: the same item in a random style (the same task instance for object re-identification and saccadic search; a same-setting instance for contour tracing). Right: another item in a second random style. Task logic, prompts, and answers do not change.

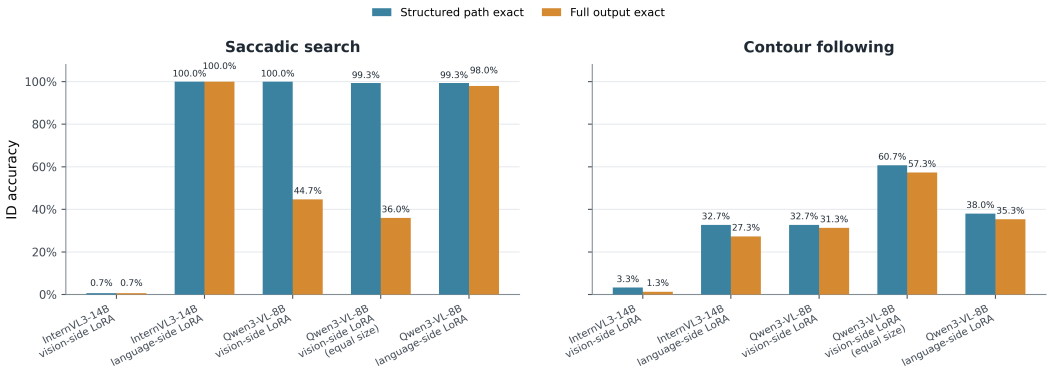


Figure 17: Full-output versus trajectory scoring for saccadic and contour process targets.

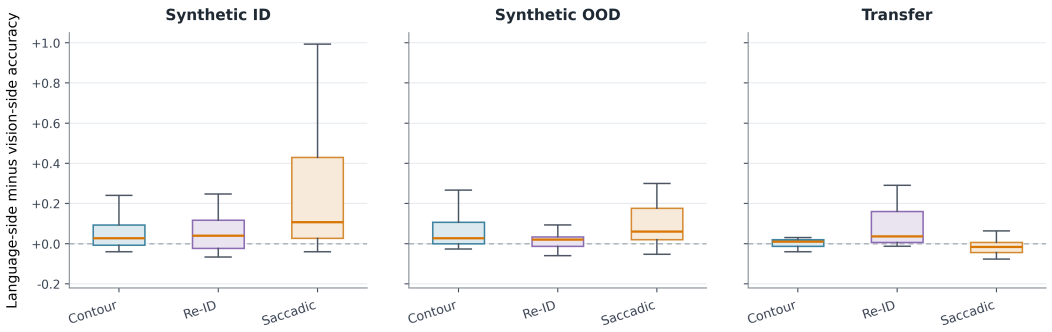


Figure 18: Language-side minus vision-side accuracy (as a fraction) for matched conditions.