ENCODED WHERE IT IS DRAWN, ABSENT WHERE IT IS READ: RELATIONAL EVIDENCE IN SCIENTIFIC DIAGRAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific diagrams make claims about processes: a food web can depict every organism correctly yet misstate energy flow if an arrow is reversed. Image generators can produce plausible diagrams with incorrect structure. We study Qwen-Image and Stable Diffusion 3.5 on controlled diagrams spanning seven scientific systems. Across layouts, a target arrow is pixel-identical in both roles, while a remote arrow determines whether its chain continues. Using the activation change caused by editing the target, we test whether the remote arrow's direction is decodable where drawn, at the target, and whether two arrows contribute when both determine the role. Under low-noise reversal edits on short chains, a Qwen-Image target-region probe achieves 99.6% accuracy when entire scientific systems are held out, versus 50.0% for a position-only baseline. Increasing the remote arrow's separation preserves the signal, whereas seven of eight redraws largely remove it. Blocking each target region's attention to its own lane's remote arrowhead nearly eliminates it in three tested layouts; blocking the other lane's, or a comparable control arrowhead, does not. In three other settings, direction is nearly perfectly decodable locally but weakly at the target. In a two-arrow task, the target readout favors the last arrow at most late blocks, although at one block pooled accuracy exceeds the best single-arrow accuracy. In Stable Diffusion 3.5 the corresponding estimate stays near zero. Accurate decoding alone therefore cannot identify which visual evidence supports a prediction or establish that all arrows defining a relation contribute.

1 INTRODUCTION

Scientific diagrams use directed relations to represent processes. In a food web, a single reversed arrow can misstate energy transfer even when every organism is depicted correctly. Systems now generate diagrams and scientific imagery (Zala et al., 2024; Lin et al., 2026). SciForma addresses structural fidelity with a FLUX.2-based diffusion transformer (Luo et al., 2026), while SciFlow-Bench documents graph errors in visually plausible generated diagrams (Zhang et al., 2026). Output-level evaluation identifies such errors; it does not establish what relational evidence is present in a generator's internal states.

Probes trained on model activations offer one way to examine that evidence. A positive result at the target region still leaves unresolved which arrows supply its evidence and whether the arrows defining the relation contribute together.

We examine this distinction in Qwen-Image (Wu et al., 2025) and Stable Diffusion 3.5 Large (Esser et al., 2024), two text-to-image diffusion transformers, using controlled diagrams spanning seven science themes. Each diagram contains two role lanes and a control lane. Across the role lanes, the target arrows connect the same icons and, in the same row across layouts, have identical pixels; a remote arrow determines whether the chain continues (Figure 1). Reversing that arrow creates a controlled counterfactual rather than an alternative account of a valid scientific process.

The design supports three distinct tests. A probe on original-image activations tests whether the remote arrow's direction is decodable where it is drawn. A probe on the target-region edit response, $\Delta H = H_{\text{edit}} - H_{\text{base}}$, tests whether distant context affects the representation near the edited target

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

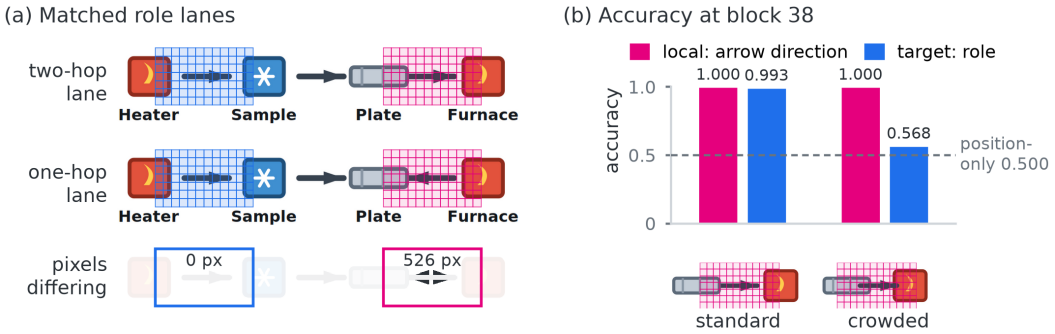

Figure 1: **Local decoding and target-region edit responses.** (a) One row of a food-web diagram in two layouts, holding the two-hop lane in one and the one-hop lane in the other. The images differ in 526 pixels, all inside the remote arrow's region (magenta), and in none of the target region (blue). (b) Under the standard rendering, local direction accuracy is 1.000 and target-role accuracy is 0.993 (distance-sweep extraction; 0.992 in Section 4.1); when the remote arrow is redrawn crowded, over the icons, they are 1.000 and 0.568. The position-only baseline is 0.500.

arrow. A separate two-arrow construction tests whether each arrow required by the role contributes to that response. Earlier versions of the test produced apparent relational signals from endpoint identity and unequal chain lengths (Table 1), and from an estimator offset under shuffled labels (Appendix A.6). These cases motivate examining the visual evidence behind the signal that remains.

This work contributes:

- ◇ We propose a controlled testbed of science-themed diagrams in which target-arrow pixels remain fixed across roles while remote arrows vary. It supports separate tests of local direction decoding, target-region edit responses, and the contribution of multiple arrows.
- ◇ We identify a relational signal in Qwen-Image under low-noise reversal edits on short chains. With entire science systems held out, the target-region probe reaches 99.6% accuracy against 50.0% for a position-only baseline. Under the original split, the signal passes all probing controls we test. In Stable Diffusion 3.5 the corresponding estimate stays near zero.
- ◇ We find that increased arrow separation preserves the Qwen-Image signal, whereas seven of eight redraws remove most of it. Blocking both target regions' attention to both remote arrowheads nearly eliminates the signal in three tested layouts; a size-matched control cut leaves most of it. Blocking only each lane's own remote arrowhead does the same, while blocking only the other lane's raises it.
- ◇ We find three settings in which remote-arrow direction is nearly perfectly decodable where drawn, while the corresponding role is weakly decodable from the target's edit response. In the two-arrow construction, the target readout favors the arrow that ends the chain at most late blocks. At one block, pooled accuracy exceeds the best single-arrow accuracy, leaving limited integration possible.

These findings concern representations and responses to controlled edits in the tested models and conditions; they do not establish which representations the models use when generating diagrams.

2 Related Work

Scientific Diagram Generation and Evaluation. DiagrammerGPT plans diagram entities, connections, and layouts before rendering (Zala et al., 2024). SciForma targets the fidelity of components, arrows, and text (Luo et al., 2026), while SciFlow-Bench evaluates generated diagrams by reconstructing their graphs from pixels (Zhang et al., 2026). Work on scientific-image synthesis likewise evaluates logical validity alongside visual quality (Lin et al., 2026). These studies assess or improve the diagrams a model produces. We examine what relational evidence is decodable from a generator's activations when it processes a controlled diagram.

Relational Information in Visual Representations. Benchmarks document compositional and spatial errors in vision-language models (Thrush et al., 2022; Yüksekgönül et al., 2023; Kamath et al., 2023) and failures to render prompted relations in text-to-image generation (Conwell & Ullman, 2022; Wu et al., 2024; Huang et al., 2023; Gokhale et al., 2022). In contrast, vision transformers fine-tuned on same–different judgments can develop distinct perceptual and relational stages (Lepori et al., 2024). For diagrams, Terashita et al. (2025) train a visual encoder to recognize edges using data designed to remove textual and positional biases. Yoshida et al. (2026) find that directed-edge information in the vision-language models they study becomes linearly decodable in language tokens, although it is not linearly separable in their vision encoders. The location of a visual token does not necessarily determine what information it carries. Some vision transformers repurpose patch tokens associated with low-information background regions for internal computation (Darcet et al., 2024). Foveated probes also recover localized binding information missed by global readouts (Michalkiewicz et al., 2026). We compare two different measurements in diffusion transformers: direction decoding at a remote arrow from original-image activations, and role decoding at a target arrow from its edit response. The two-arrow diagrams further test whether both arrows required by a role contribute to the target readout.

Probes and Their Controls. A probe establishes that a property is accessible to a specified readout; accessibility need not imply relevance to the model's task (Belinkov, 2022; Ravichander et al., 2021). Successful probes have been interpreted as evidence of emergent world representations (Li et al., 2023) and scene structure in diffusion models (Chen et al., 2023). Our ΔS follows conditional probing by measuring what the target-region activations add beyond position (Hewitt et al., 2021). Control tasks and label shuffling check selectivity and estimator behavior; information- and description-length criteria assess recoverability and coding cost (Hewitt & Liang, 2019; Pimentel et al., 2020; Voita & Titov, 2020).

Causal Analysis of Diffusion Transformers. Wang et al. (2026) trace circuits for generating prompted spatial relations between two objects in DiTs trained from scratch, comparing random text embeddings with a pretrained T5 text encoder. We study pretrained generators processing diagrams supplied as images, and ask whether access to remote arrowheads is required for a target-region edit-response signal. Causal mediation (Vig et al., 2020), activation patching (Meng et al., 2022), and attention knockout (Geva et al., 2023) provide precedents for intervening on model computations. In text-to-image models, related work localizes and edits stored visual attributes (Basu et al., 2024); Ge et al. (2026) use attention knockout and activation patching to trace how reference-image content flows through image and text tokens in FLUX.2. Our joint cut blocks both role lanes' target regions from attending to both remote arrowheads. Its effect establishes dependence under that intervention; the cut alone does not isolate the supplying lane or reconstruct the unintervened information path.

3 METHOD

We ask three questions about a target arrow's *role*, whether the chain continues past the node it points into. Is the direction of the remote arrow that decides the role decodable where that arrow is drawn? Does it influence the target region's response when the target arrow is edited? And when the role depends on two remote arrows, does each contribute to that response? This section defines the corpus, the readouts behind each question, and the checks an estimate must pass before it is read.

3.1 MODELS AND ACTIVATION EXTRACTION

We study Qwen-Image (20B parameters, 60 transformer blocks; Wu et al., 2025) and Stable Diffusion 3.5 Large (8B parameters, 38 blocks; Esser et al., 2024), open-weight diffusion transformers whose internals we can read and change. Each 1024×1024 image becomes a 64×64 grid of image tokens. We add noise at a *renoising strength* s, which sets the point in the denoising trajectory the model is conditioned on, run the model once, and record the output of selected blocks. Qwen-Image results use $s = 0.2$ and Stable Diffusion 3.5 results $s = 0.4$ unless stated; both models were also run at the other strength. We record four blocks spread through each model's depth and four more between the last two; the late Qwen-Image blocks we report are 35, 38, 41, 44 and 47 (the blocks recorded for each family are listed in Appendices A.2 and A.3).

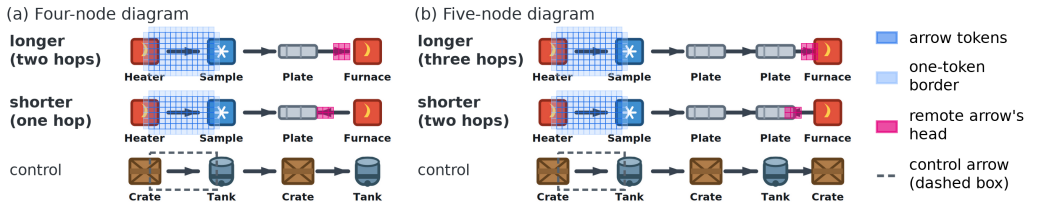

Figure 2: **Target arrow, remote arrow and readout region.** Blank space between lanes removed. **(a)** A four-node diagram: the two upper lanes are identical but for the remote arrow, which points onward in the two-hop lane and back in the one-hop lane. **(b)** The five-node version, repeating that contrast one hop further. Blue: the tokens covering the target arrow (Heater→Sample). Light blue: the one-token border. Magenta: the head of the remote arrow, which decides the role. Dashed: the control arrow of the bottom lane.

3.2 Controlled Diagram Corpus

Real diagrams rarely contain two arrows that look the same but play different roles, so we generate our own: 112 base diagrams from seven scientific systems, sixteen variants each, each drawn in all six orderings of its three lanes and run with two noise seeds (Figure 2; Appendix A.1). Two lanes carry a target arrow, matched on every visible property, with any mismatch treated as an error (none occurs) (Appendix A.6); they differ only in the direction of a later arrow, the *remote arrow*, which decides how far the chain runs on. Half the diagrams give a lane four nodes and half five, the contrast repeating one hop further in the longer ones, so the role is always which lane continues farther, never the hop count itself; the analyses that trace the signal to its source use the 56 four-node diagrams. The third lane is a control lane in icons used nowhere else, whose *control arrow* gives the probe an easy, known difference for checking that the measurement works.

3.3 Paired Edits and the Readout Region

Each edit deletes or reverses exactly one arrow—a target arrow or the control arrow—and the image is redrawn from the edited graph rather than painted over (Appendix A.2). With prompt, noise seed, model, strength and block fixed, we take the per-token difference between edited and original activations, $\Delta H_u^{(\ell)} = H_{\text{edit},u}^{(\ell)} - H_{\text{base},u}^{(\ell)}$ at block ℓ. For an edited arrow i, let $\mathcal{B}_i$ be the image tokens covering its bounding box and $\mathcal{R}_i$ the one-token border around them, which absorbs misalignment between image and token grid. Together they form the *readout region*; the probe sees nothing outside it, and averages the two parts separately and concatenates them:

$$z_i^{(\ell)} = \left[\frac{1}{|\mathcal{B}_i|} \sum_{u \in \mathcal{B}_i} \Delta H_u^{(\ell)} \; ; \; \frac{1}{|\mathcal{R}_i|} \sum_{u \in \mathcal{R}_i} \Delta H_u^{(\ell)} \right]. \tag{1}$$

For target-arrow edits the readout region never overlaps the remote arrow's head, checked for every lane in every layout: from the center of the readout region to the arrowhead's centroid on the 1024 px canvas, the head sits 422–459 px away in four-node diagrams and 602–637 px in five-node ones. The body uses that convention throughout; the appendices measure from the target arrow, about 159 px shorter, and say so where they first do. Information about the remote arrow found in $z_i^{(\ell)}$ cannot have been read off pixels inside the readout region.

3.4 Role Probes and the Information Score

For each edited target arrow i, let y_i be its role and p_i its position: its bounding box's center, size and aspect ratio, the cell of an 8×8 grid its center falls in, and the share of the token grid it covers. On the same examples and folds we fit a position-only probe $q_{\text{pos}}(y_i \mid p_i)$ and one that also sees the readout, $q_{\text{act}}(y_i \mid p_i, z_i^{(\ell)})$. The score is how many bits the activations save, $\Delta S = S(Y \mid P) - S(Y \mid P, Z^{(\ell)})$, where S is the mean held-out negative $\log_2$ probability of the true role, per edit. This is a conditional probe with position as the baseline (Hewitt et al., 2021): a positive ΔS means the activations carry information about role that position does not, but says

nothing about where it came from. We report accuracy beside every estimate. The probe's settings are fixed once for every role probe (Appendix A.4); the three supporting readouts named at the end of Section 3.5 are calibrated separately. One probe is fitted across all training diagrams, because separate probes per diagram gave biased estimates—the displaced estimator of Section 4.2—and pooling removes the bias (Appendix A.6). Folds are grouped by base diagram, so every layout and seed of a held-out diagram is scored by a probe that never saw them; ΔS is computed per held-out diagram and then averaged. Every base diagram, however, has a *twin* that renders to the same six images, since putting the role on the other lane and swapping the two lanes draws the same picture, and these folds left the twin in training for 44 of the 56 four-node diagrams. We therefore repeat the main results, after the plan, with twins in one fold and with each of the seven scientific systems held out in turn (Appendix A.20). Key analyses are repeated with a small multilayer perceptron whose penalty is set on the shuffled-label control (Appendix A.4).

3.5 When a Result Can Be Trusted

Confidence intervals are 90% percentile bootstrap intervals over base diagrams, so renderings of the same graph are not counted as independent; twins are, which narrows them by about $\sqrt{2}$ (Appendix A.20). Before reading any role estimate we ask three questions (Appendix A.4).

Does the measurement work? With the same features, folds and probe, the activations must tell a target-arrow edit from a control-arrow edit, with the interval above 0.10 bits and accuracy above the position-only 0.667 implied by two target-arrow edits and one control-arrow edit per diagram. An estimate that fails this *validity check* is unreadable, which is not evidence that role is absent.

Is zero really zero? We rerun the role probe with labels shuffled within each base diagram and read the offset that remains. An offset of at most 0.010 bits is clean; a larger one is tolerated, and reported beside the estimate, only if it is at most a tenth of $|\Delta S|$, and otherwise leaves the estimate unreadable in bits. We judge the offset by its size, not by whether its interval straddles zero (Appendix A.18).

Is a small result small enough? A result is *negligible* only if its whole interval lies inside a margin fixed in advance (Lakens et al., 2018), $[-\delta, +\delta]$ with $\delta = 0.10$ bits, and *inconclusive* if its interval crosses that margin.

Control-task selectivity (Hewitt & Liang, 2019) and online-code description length (Voita & Titov, 2020) are run on every apparent signal and reported, never used as gates: neither says where a signal comes from (Appendix A.5).

Three analyses support the main results but are never the only basis for a claim: a pairwise endpoint readout, calibrated on its own shuffled-label control (Appendix A.14); an exploratory activation transplant (Appendix A.15); and 349 real diagrams from AI2D-RST (Kembhavi et al., 2016; Hiippala et al., 2021) (Appendix A.16). Analyses *specified in advance* were written into an append-only plan file before the run they govern (Appendix A.17); those added afterwards are marked where they appear. Every claim is limited to what our probes read from these two models, at these noise levels and blocks, on these diagrams and readout regions.

3.6 Locating the Evidence Behind the Target Readout

For the one signal that passes these checks—Qwen-Image, reversal edits, $s = 0.2$, four-node diagrams—we take two steps, each holding the graph fixed and changing one thing, and then build diagrams in which the candidate sources come apart.

Redrawing the remote arrow. Roles, objects, arrow directions and the target arrow's exact pixels stay the same; only the remote arrow is redrawn, in versions from moving it about 150 pixels farther away to covering its head or stretching its shaft (Appendix A.7), each probed at block 38. Because two separately rendered images can differ in ways we did not intend, this points to a likely source rather than settling it.

Cutting its route. Here the image stays the same and we change what the model can attend to: at every block, the tokens covering both lanes' target arrows—each edit's readout region without its one-token border, since the cut is applied once per layout rather than per edit—are prevented from attending to the tokens covering the remote arrows' heads, with all other attention unchanged

(Appendix A.8). A *size-matched cut* blocks a similar number of tokens over the control arrow's head instead. Two *lane-separated cuts*, added after the plan, block each lane's target arrow from only its own lane's remote arrow's head, or only the other lane's. Because blocking renormalizes attention, all cuts are compared with an empty mask through the same code rather than with an unmasked run. The cuts run in both models, Stable Diffusion 3.5 at its usual $s = 0.4$. These runs use three of the six layouts and are compared only with each other.

Families that change the layout or the rule. Five further families, each specified in advance, change the lane layout, the rule that decides the role, or both, and each is defined where its result is reported. *Two-arrow* separates a readout that needs one arrow from one that needs both (Section 4.5; Appendix A.9); *shifted* and *bent* move or fold the lanes (Appendix A.11); *branching* puts two arrows at the same distance (Appendix A.12); and *loop-versus-skip* asks whether the chain closes into a loop (Appendix A.13).

4 RESULTS

Every score is ΔS, read only when the checks of Section 3.5 pass (full tables: Appendix A.21).

4.1 AN APPARENT ROLE SIGNAL IN ONE CONDITION

Across models, edit types, noise levels and blocks, almost every cell of the planned grid sits near zero. One does not: Qwen-Image, reversal edits, $s = 0.2$, where a signal appears in the later blocks and nowhere earlier. It first showed at block 47; a denser sweep of blocks, specified in advance (Appendix A.17), puts the peak at block 38, $+0.381$ bits (CI_{90} $[+0.328, +0.433]$, accuracy 0.794 against a position-only 0.500, shuffle -0.001). That peak is over all 112 diagrams and six layouts; the three-layout attention-cut runs of Section 4.4 peak at block 47, so blocks are compared only within one set. Deleting the arrow instead of reversing it, raising the noise level, or switching to Stable Diffusion 3.5 leaves every readable estimate inside the margin (two early deletion cells fail validity; Table 19).

Nearly all of the signal lies in the four-node diagrams. With each science-themed system held out in turn, the target-region probe reaches accuracy 0.996 against a position-only 0.500 at block 38, at least 0.969 in every system, and 0.993 when the block is also chosen without the held-out system; with twin pairs kept in one fold it is $+0.925$ bits (CI_{90} $[+0.898, +0.950]$, accuracy 0.990). The folds fixed in advance, grouped by base diagram alone, give $+0.935$ (CI_{90} $[+0.918, +0.951]$, accuracy 0.992) against $+0.017$ (0.562) for the five-node diagrams on the same probe and folds, both with a clean shuffle; the split by diagram size was run after the signal was found. The pooled $+0.381$ sits below the mean of the halves because one probe fitted on both must serve conditions that disagree. The rest of this section uses the 56 four-node diagrams at block 38 under those folds; Section 4.7 returns to the five-node null.

4.2 WHAT EACH CONTROL CAN AND CANNOT ESTABLISH

Three other apparent signals in this project looked as convincing as this one and had different causes: endpoint identity, chain length and a displaced estimator. Table 1 scores the first two beside the remote-arrow signal by the same checks (Appendix A.6); the label shuffle exposed the displaced estimator, whose whole result was an offset. No standard control located a source, and the remote-arrow signal, whose information comes from outside the readout region, passed every one.

4.3 THE SIGNAL FOLLOWS HOW THE REMOTE ARROW IS DRAWN, NOT WHERE IT IS

Holding the graph, the role labels and the target arrow's pixels fixed, we redrew only the remote arrow (Figure 10a; Appendix A.7). Distance costs nothing: moving its head in three steps from the corpus's 422–459 px out to about 576–612 px leaves the estimate between $+0.792$ and $+0.969$ bits at accuracies of 0.952 to 1.000, with a clean shuffle at every step and no decay.

Drawing decides it. Seven of the eight redraws that leave the graph and labels untouched collapse the signal at block 38, in three kinds. Covering the arrowhead brings $+0.935$ down to $+0.071$ (accuracy

Table 1: What each level of checking establishes for the three apparent signals in the stimuli. Each row adds a check; values are ΔS in bits, about which edge was edited for endpoint identity and about role otherwise. Shuffle verdicts are rescored under the magnitude rule. Bold marks the level that resolved each case. Intervention values use three of the six layouts.

	Endpoint identity Qwen-Image, which edge, $+0.539$	Chain length SD3.5 MLP, $+0.553$	Remote arrow Qwen-Image, $+0.935$
Score alone	Apparent signal	Apparent signal	Apparent signal
+ validity check	Not applicable (no control edge)	Passes	Passes
+ label shuffle	Offset -0.216: estimate unreadable, cause not named	Clean (-0.0007)	Clean ($+0.001$)
+ selectivity, MDL	Both flag (-0.076, -0.611); cause not named	Both flag ($+0.003$, -0.041); cause not named	Both clean ($+0.343$, $+0.786$); source not located
+ redesign	**Identifies shortcut** ($\rightarrow -0.001$)	**Identifies shortcut** ($\rightarrow +0.00002$)	Needs the standard drawing ($+0.008$ to $+0.094$); cause not yet identified
+ intervention	—	—	**Establishes dependence** on arrowhead access ($+0.730 \rightarrow +0.005$; size-matched cut $+0.636$)
+ dissociation	—	—	**Identifies one arrow**: role split across two arrows, the probe follows the final one (block 44; $+0.96$ against $+0.04$)

0.612), and covering 35% or 68% of it is no gentler ($+0.094$, $+0.082$). Stretching the shaft, so that the head returns to the near distance while the drawn line matches the farthest condition, gives $+0.019$ (0.547). Tightening the gap the arrow is drawn in gives $+0.016$ (0.561), $+0.021$ with the arrow drawn on top of the icons, and $+0.008$ with that gap pushed out to the far end of the sweep. The eighth redraw keeps this from being circular: sliding the neighboring icon the same distance *away* from the arrowhead, leaving the head fully visible, costs nothing ($+0.934$, accuracy 0.995). Every shuffle is clean and the position-only baseline stays at 0.500.

Tightening the gap puts the arrow's head 238 px from the target arrow against the standard 263 px (both measured as in Appendix A.7), and the variant pushed out to 443 px, farther than any swept step, is equally null: crowding, not distance, is what they share. Read at its own region, the crowded arrow's direction is still recoverable, at accuracy 1.000 at every late block, so crowding weakens the target readout rather than the local encoding. Separately rendered images can differ in unintended ways, so this narrows the suspects without convicting one.

4.4 Cutting Attention to the Arrowhead Removes the Signal

Redrawing changes the picture; the intervention does not. We cut the attention of both lanes' target arrows—the readout region without its border—to the remote arrowheads (Figure 10b,c; Appendix A.8; plan entry written after the run, Table 16). These runs use three of the six layouts, so the reference is $+0.730$ rather than $+0.935$; unmasked, those layouts give $+0.7299$ against the empty mask's $+0.7301$, so the masking code alone changes nothing.

In Qwen-Image the cut ends the signal at every late block: at block 38 it takes $+0.730$ to $+0.005$, and across blocks 35 to 47 the unmasked estimates run from $+0.174$ to $+0.943$ while every masked one lies between -0.005 and $+0.005$. The size-matched cut, 12 tokens against 15, leaves $+0.636$ at block 38, and validity passes in all three conditions (0.951 to 0.980), so the cut removes the signal, not the measurement. In Stable Diffusion 3.5 every value is already inside the margin, so we read accuracy against the position-only 0.464 of these layouts: at the four blocks where the unmasked probe reads most (24, 26, 28 and 30) it falls from 0.55–0.57 to 0.49–0.51 under the cut and stays at 0.55–0.58 under the size-matched one.

The lane-separated cuts, added after the plan, show whose arrowhead the readout needs. Blocking each target arrow from only its own lane's remote arrow's head ends the signal as the joint cut does, -0.004 to $+0.001$ bits at blocks 35 to 47 (-0.003, accuracy 0.496, at block 38). Blocking it from only the other lane's instead raises the signal at every late block, to $+0.909$ (0.985) at block 38, where the size-matched cut leaves $+0.636$; validity passes under every cut (Appendix A.8). The readout needs its own lane's remote arrow, not the other's; why the other cut raises it we did not test.

4.5 Is It the Role, or One Arrow?

In this corpus the remote arrow's direction and the target arrow's role are the same variable, so a probe reading either makes identical predictions. The *two-arrow* family separates them: the role depends on two arrows, and the two lanes of each diagram share one of the two directions, so one subset is separable only by the last arrow and the other only by the next-to-last (Figure 10d; Appendix A.9). A representation of the role must separate both subsets; a readout of one arrow separates one.

The probe reads one arrow. Of the five late blocks (35, 38, 41, 44 and 47), pairs that differ in the last arrow are separated at three: $+0.959$ bits at 1.000 accuracy at block 44, $+0.591$ at block 35 with a label-shuffle flag, and $+0.368$ at block 38, clean. Block 41 is inside the margin, at $+0.040$; block 47, at $+0.087$ (CI_{90} $[+0.056, +0.122]$), is inconclusive, and with twins in one fold it clears the margin, $+0.292$ $[+0.180, +0.403]$. The separation is not monotone in depth, and we do not explain that. Pairs that differ in the next-to-last arrow never clear the margin, at $+0.011$ to $+0.076$ bits (at most $+0.091$ with twins in one fold), though not at chance: their accuracy is 0.56 to 0.62 at the late blocks against 0.51 to 0.52 at blocks 9 to 32, a trace below the margin.

The arrow the probe follows is the farther one, at about 596–633 px against 391–428, but four-node diagrams whose single remote arrow is pushed out to 596–633 px read at $+0.93$ to $+0.99$ bits and accuracy 0.995 to 1.000, so distance is not the explanation. What separates the two is that one ends the lane—a fact about the chain, not about drawing, which Section 5 takes up. One caveat: at block 44 the trace lifts the estimate pooled over both subsets to $+0.465$ bits at 0.833 accuracy, above the 0.311 bits and 0.750 that a probe reading only one arrow can reach, so some integration there is not excluded; we read the family by the per-subset contrast it was built to make (Appendix A.9). Read at each arrow's own region instead, both directions are recoverable at accuracy 1.000 against 0.500 in every cell but one (0.996), the next-to-last included, though no cell there has a clean shuffle and the claim rests on accuracy (Appendix A.10). What fails for the earlier arrow is arrival, not encoding, which renames the asymmetry rather than explaining it.

4.6 Shifted, Bent and Branching

Three further families move the layout or change the rule, each specified in advance (Appendices A.11 and A.12). *Shifted* translates each lane horizontally while the geometry inside it stays pixel-identical. The probe fitted on the original corpus transfers without refitting at 0.916 accuracy (CI_{90} $[0.884, 0.944]$), against the 0.992 it reaches on its own source, so absolute position is not what it learned; the attention cut reproduces there, taking accuracy from 0.981 to 0.510 while a size-matched cut leaves 0.984. *Bent* folds the chain onto two sub-rows and removes the signal both under transfer (0.560) and under refit (0.548, $\Delta S = +0.007$ bits). We do not claim the bend is the cause, since the fold also enlarges the readout region by about 5%. *Branching* puts two mirror-image arrows at equal distance, both of them ending the chain, so neither distance nor position in the chain picks one out—the two explanations the two-arrow family could not separate—and the probe separates *neither* subset, at accuracies of 0.490 to 0.521 with validity passing throughout. Each branch direction is nonetheless recoverable at its own region at accuracy 1.000: the directions are encoded where they are drawn and do not arrive at the readout region.

4.7 What Remains

With the shortcuts removed, most conditions carry no recoverable role information, and not because the instrument fails: on the same features and folds the probe reads the easy target-versus-control difference in both models, passing the validity check, while role sits at chance. The loop-versus-skip family, which asks whether the chain downstream closes into a loop or bypasses it, gives -0.001 to $+0.002$ bits at 0.50 to 0.52 accuracy, with validity at 0.81–0.88 and a clean shuffle at every block (Appendix A.13). On 349 real diagrams from AI2D-RST the activations locate a deleted arrow precisely yet cannot say which of two similar arrows went missing (Appendix A.16).

The five-node null is not a fact about hop count. Those diagrams draw the remote arrow into a narrow gap, the crowding that nulls the four-node signal in Section 4.3, and the two agree at $+0.017$ and $+0.016$ bits. The two-arrow diagrams are also five-node but drawn at the four-node corpus's

widest spacing, and there the last arrow reaches $+0.959$ bits; that family also changes the role rule, so it supports the crowding reading rather than settling it.

Nor are these nulls an under-confident probe scoring few bits while separating the classes. Under a rule specified in advance, 147 of the estimates inside the margin could be refit and two were flagged, both endpoint readouts of Appendix A.14, where we had predicted that failure; no other audited estimate was flagged, and the partial cuts and graded redraws could not be refit (Appendix A.19).

5 Discussion

The signal that survived every standard control is dominated by one arrow's drawn direction. That is a shortcut with respect to our question, which was about the chain, but not a meaningless one: the remote arrow's direction is part of the graph, and in our original diagrams it determines the role. What the probe mostly does not do is compose; where the role takes two arrows, it follows one.

Which one it follows raises a question of its own. In *two-arrow* the probe follows the arrow that ends the chain—a structural fact, which sits awkwardly beside a headline about drawing. *Branching* shows that ending the chain is not sufficient: both branch arrows end it, each is recoverable at accuracy 1.000 where it is drawn, and neither reaches the readout region. Whether it is necessary we cannot say, because our designs do not separate an arrow's terminal position from how the end of a lane is drawn—the empty space past it, for instance.

One placement fact separates the families that keep the signal from those that lose it; we found it after the fact and report it as an observation. The signal survives wherever the remote arrow sits on the target arrow's row: in the original corpus, in *shifted*, whose lanes move but stay flat, and in *two-arrow*, whose nodes are collinear. It disappears where the row breaks: *bent* separates the sub-rows by 110 px and *branching* offsets each tip by 56 px, about seven and three and a half tokens, while a horizontal move of about 150 px, some nine tokens, costs nothing (Section 4.3). The readout region may draw on a narrow horizontal band rather than on distance in any direction. That accounts for the families that lose the signal but not for the *two-arrow* asymmetry, and both off-row families change something else too—*bent* enlarges the readout region by about 5%, *branching* replaces the role rule—so the pattern is consistent with the evidence rather than established by it. Testing it needs a family that moves the remote arrow off the row with the readout region and the role rule held fixed.

The problem is not specific to diagrams. Any probe that reads a region—image patches in a vision transformer, token positions in a language model—treats its activations as evidence about local content, and attention can bring in information from anywhere in the input, in a form that passes selectivity, description length and shuffled-label checks. Probing studies of board states (Li et al., 2023) and scene structure (Chen et al., 2023) go beyond standard controls by intervening on the representation they probe. That shows the model uses it, not which part of the input it was built from or that it encodes the intended property rather than a correlate. The steps of Section 3.6 address that second question and carry over with the stimuli adapted.

Our claims are bounded by what the probes read. The signal appears in one condition of the planned grid, nearly all of it in the four-node diagrams—a split we made after seeing the result—so the source tracing rests on 56 diagrams and, for the strong signal, on one model, at these blocks and noise levels. We isolate no single property of the drawing, and cannot rule out that the model computes role only for arrows drawn as in our original corpus, in which case our redraws disrupt that computation rather than expose a shortcut. We do not show that either model uses this information when it generates an image, and a probe that fails to read a property does not establish that the model lacks it.

6 Conclusion

Validating a probe and identifying the source of its signal are different problems: the first asks whether a property can be read out, the second which part of the input the reading rests on. In Qwen-Image every standard control we ran accepted a role signal dominated by one arrow's drawn direction and carried to the readout region by attention, and the same tools found directions encoded where they are drawn that do not reach it, a failure no standard control can tell from absence. Redraw, cut and dissociate exposed both; we offer the procedure for any probe whose target, like a relation, is determined outside the region it reads.

AI USE STATEMENT

We used an LLM-based assistant to implement stimulus generation, activation extraction, probing, attention masking, and analysis; generate the synthetic corpus and edited AI2D-RST stimuli; propose and refine hypotheses; advise on experimental designs and controls; and help interpret results. It also assisted with code, figures, tables, literature checks, and drafting and editing the paper, including its structure and title. We did not use it for mathematical claims, proofs, or translation; qualitative and thematic analysis is not applicable.

The authors set the research question, selected the experiments, and approved the interpretations. Decisions described as specified in advance were recorded before the relevant experiments in an append-only plan (Appendix A.17). We checked reported numbers against saved outputs, tested AI-assisted analysis instruments on known-answer cases, verified references, and documented failed checks in Appendix A.18.

REPRODUCIBILITY STATEMENT

Section 3 specifies the models, image resolution, renoising strengths, extracted blocks, readout region, positional features, cross-validation scheme and interpretation criteria. We use the public checkpoints `Qwen/Qwen-Image-2512` and `stabilityai/stable-diffusion-3.5-large`, a 40-step noise schedule, and renoising noise seeds 0 and 1; confidence intervals are 90% cluster-bootstrap intervals over base diagrams, and the equivalence margin is ± 0.10 bits. Appendix A.1 describes how the 112 base diagrams are generated, including the seven scientific systems, the sixteen variants, the matching rules, the six layouts and the edits. Appendix A.2 traces a diagram through extraction and counts the 220,718 forward passes of the main analyses. Appendices A.7 and A.8 specify the redrawn stimuli and the attention masks; the tokens removed in partial masking are drawn at random from a seed fixed per diagram and layout. Appendix A.16 gives the selection and editing procedure for the AI2D-RST diagrams. Appendix A.14 gives the design, gates and stopping rule of the pairwise endpoint readout, all specified in advance, together with the post-hoc calibration diagnostic run on its saved output. Appendix A.3 reports the hardware, software versions, runtime and storage.

The calibration audit of Section 4.7 is pre-registered, in the sense Section 3.5 calls specified in advance, rather than post hoc. Its flag rule, its scope, and the four estimates for which we predicted flags were written into the plan file and frozen before it ran, and it was run once. We will release the frozen rule document and the inventory of estimates inside the margin together with their SHA-256 digests, which the audit script recomputes at run time and embeds in its output, so the reported flags can be checked against the rule that produced them. We will also release what was seen before the freeze: sixteen estimates were visible while a refit path was being validated, with no threshold or formula changed afterwards.

Activations were not retained for every earlier run. Where the stimuli and parameters allowed it we re-extracted them, leaving the originals in place; against the surviving item lists these re-extractions give identical row sets and reproduce every stored estimate to five decimal places on 64 of 64 cells, so the audit refits the same rows as the reported analyses rather than an approximation of them. Of the estimates that fall inside the margin, 147 could be refit on this basis. The rest are reported as unaudited, and are not counted as having passed the check; for the earliest runs the activations cannot be recovered at all, and Appendix A.19 counts them by group.

We will release the stimulus-generation, extraction, probing and masking code, the stimulus specifications and the saved result files upon publication. Rather than redistribute edited AI2D images, which would have to carry its share-alike licence, we will release the script that rebuilds the edited stimuli from the original AI2D and AI2D-RST downloads.

ETHICS STATEMENT

This study involves no human participants and no personal data. The main corpus consists of synthetic diagrams drawn by our own code from simple icons and text labels. The real-diagram analysis uses AI2D (Kembhavi et al., 2016), released by the Allen Institute for AI under CC BY-SA 4.0, and

the AI2D-RST annotations (Hiippala et al., 2021), released under CC BY 4.0; we show a small number of diagrams as examples. Qwen-Image and Stable Diffusion 3.5 Large are used under their respective licenses (Apache 2.0 and the Stability AI Community License), and we do not release generated images. The work analyzes what these models represent rather than enabling new capabilities, and we see no direct route to harmful use. Its main risk is over-interpretation, so we state throughout that our results concern decodability under specific probes and do not show that the models understand or use graph structure.

REFERENCES

Samyadeep Basu, Nanxuan Zhao, Vlad I. Morariu, Soheil Feizi, and Varun Manjunatha. Localizing and editing knowledge in text-to-image generative models. In *ICLR*, 2024.

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

Yida Chen, Fernanda Viégas, and Martin Wattenberg. Beyond surface statistics: Scene representations in a latent diffusion model. In *NeurIPS Workshop on Diffusion Models*, 2023.

Colin Conwell and Tomer D. Ullman. Testing relational understanding in text-guided image generation. *arXiv preprint arXiv:2208.00005*, 2022.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

Chris Ge, Rohit Gandikota, Antonio Torralba, and Tamar Rott Shaham. Vision-language binding in in-context image generation. *arXiv preprint arXiv:2605.24624*, 2026.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In *EMNLP*, 2023.

Tejas Gokhale, Hamid Palangi, Besmira Nushi, Vibhav Vineet, Eric Horvitz, Ece Kamar, Chitta Baral, and Yezhou Yang. Benchmarking spatial relationships in text-to-image generation. *arXiv preprint arXiv:2212.10015*, 2022.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *EMNLP-IJCNLP*, 2019.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher D. Manning. Conditional probing: Measuring usable information beyond a baseline. In *EMNLP*, 2021.

Tuomo Hiippala, Malihe Alikhani, Jonas Haverinen, Timo Kalliokoski, Evanfiya Logacheva, Serafina Orekhova, Aino Tuomainen, Matthew Stone, and John A. Bateman. AI2D-RST: A multimodal corpus of 1000 primary school science diagrams. *Language Resources and Evaluation*, 55 (3):661–688, 2021.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS Datasets and Benchmarks Track*, 2023.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? Investigating their struggle with spatial reasoning. In *EMNLP*, 2023.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

Daniël Lakens, Anne M. Scheel, and Peder M. Isager. Equivalence testing for psychological research: A tutorial. *Advances in Methods and Practices in Psychological Science*, 1(2):259–269, 2018.

Michael A. Lepori, Alexa R. Tartaglini, Wai Keen Vong, Thomas Serre, Brenden M. Lake, and Ellie Pavlick. Beyond the doors of perception: Vision transformers represent relations between objects. In *NeurIPS*, 2024.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *ICLR*, 2023.

Honglin Lin, Chonghan Qin, Zheng Liu, Qizhi Pei, Yu Li, Zhanping Zhong, Xin Gao, Wei Li, Wentao Zhang, Yanfeng Wang, Conghui He, and Lijun Wu. Scientific image synthesis: Benchmarking, methodologies, and downstream utility. In *ECCV*, 2026.

Yuxuan Luo, Peng Zhang, Xinjie Zhang, Xun Guo, Zhouhui Lian, and Yan Lu. SciForma: Structure-faithful generation of scientific diagrams. *arXiv preprint arXiv:2607.18091*, 2026.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

Mateusz Michalkiewicz, Mahsa Baktashmotlagh, and Guha Balakrishnan. Foveated probes recover localized binding information in vision foundation models. *arXiv preprint arXiv:2608.00726*, 2026.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *ACL*, 2020.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *EACL*, 2021.

Naoyuki Terashita, Yusuke Tozaki, Hideaki Omote, Congkha Nguyen, Ryosuke Nakamoto, Yuta Koreeda, and Hiroaki Ozaki. Can visual encoder learn to see arrows? In *CVPR Workshop on Visual Concepts*, 2025.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Investigating gender bias in language models using causal mediation analysis. In *NeurIPS*, 2020.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *EMNLP*, 2020.

Binxu Wang, Jingxuan Fan, and Xu Pan. Circuit mechanisms for spatial relation generation in diffusion transformers. In *CVPR*, 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Yinwei Wu, Xingyi Yang, and Xinchao Wang. Relation rectification in diffusion model. In *CVPR*, 2024.

Haruto Yoshida, Keito Kudo, Yoichi Aoki, Ryota Tanaka, Itsumi Saito, Keisuke Sakaguchi, and Kentaro Inui. Nodes are early, edges are late: Probing diagram representations in large vision-language models. *arXiv preprint arXiv:2603.02865*, 2026.

Mert Yüksekgönül, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. DiagrammerGPT: Generating open-domain, open-platform diagrams via LLM planning. In *COLM*, 2024.

Tong Zhang, Honglin Lin, Zhou Liu, Chong Chen, and Wentao Zhang. SciFlow-Bench: Evaluating structure-aware scientific diagram generation via inverse parsing. In *ACL*, 2026.

A Appendix

Appendix roadmap. This appendix follows the order of Section 3. Part I is the corpus and the extraction, Part II the probe and the checks, Part III the two experiments that locate the source of the signal, Part IV the five further diagram families, Part V the supporting analyses, and Part VI the record of what was specified in advance, the calibration audit and the full tables. The main text stands on its own; this is for a reader checking something specific.

Part I Corpus and Extraction

How the diagrams are built, how a diagram becomes activations, and what that cost. Appendices A.1–A.3.

A.1 Seven Scientific Systems

The corpus has 112 base diagrams: seven scientific systems, sixteen variants each. Every diagram is drawn by code from a graph description on a 1024×1024 canvas, with nodes as icons labeled underneath and edges as arrows. Each has three lanes, two of which carry a target arrow — the *role lanes* — and one control lane. Figure 3 shows one diagram from each system.

Systems and relations. Each system describes one scientific process, such as heat moving between objects or energy moving through a food web. Within a system every arrow means the same thing, so the relation never changes; only the arrow's position in the graph does. The relation is not written on the image. Table 2 lists the entities and relation for each system. A few icons are reused under a different label — the sun icon is labeled *Lamp* in the light-response system — but the two role lanes of a diagram always use the same icons and labels.

Table 2: The seven scientific systems. Each role lane is the chain *source* $\rightarrow$ *target* $\rightarrow$ *next hop* $\rightarrow$ *tail*, and the target arrow runs from source to target. The control lane uses the same two entities in every system.

System	Domain	Relation	Target arrow	Next hop / tail
Heating	Thermal physics	causes temperature increase	Heater $\rightarrow$ Sample	Plate / Furnace
Light response	Plant biology	causes growth response	Lamp $\rightarrow$ Seedling	Plant / Seeds
Food web	Ecology	transfers energy to	Grass $\rightarrow$ Grasshopper	Frog / Hawk
Matter cycle	Earth systems	transfers matter to	Cloud $\rightarrow$ Rain	Ocean / Sun
Circuit	Electricity	transfers electrical energy	Battery $\rightarrow$ Bulb	Switch / Cell
Growth chain	Life cycles	develops into	Seed $\rightarrow$ Sprout	Plant / Seeds
Feedback	Control systems	sends information to	Sensor $\rightarrow$ Controller	Actuator / Indicator

Two lanes that differ in one arrow. Writing S for the source node, T the target, M the next hop and X the tail, the four-node contrast of Section 3 is

$$\text{two-hop: } S \rightarrow T \rightarrow M \rightarrow X, \qquad \text{one-hop: } S \rightarrow T \rightarrow M \leftarrow X,$$

with the target arrow $S \rightarrow T$ identical in both. The five-node version repeats the contrast one hop further, three hops against two, so these two names hold for four-node lanes only. In the released code the two roles are named `extended` and `terminal`.

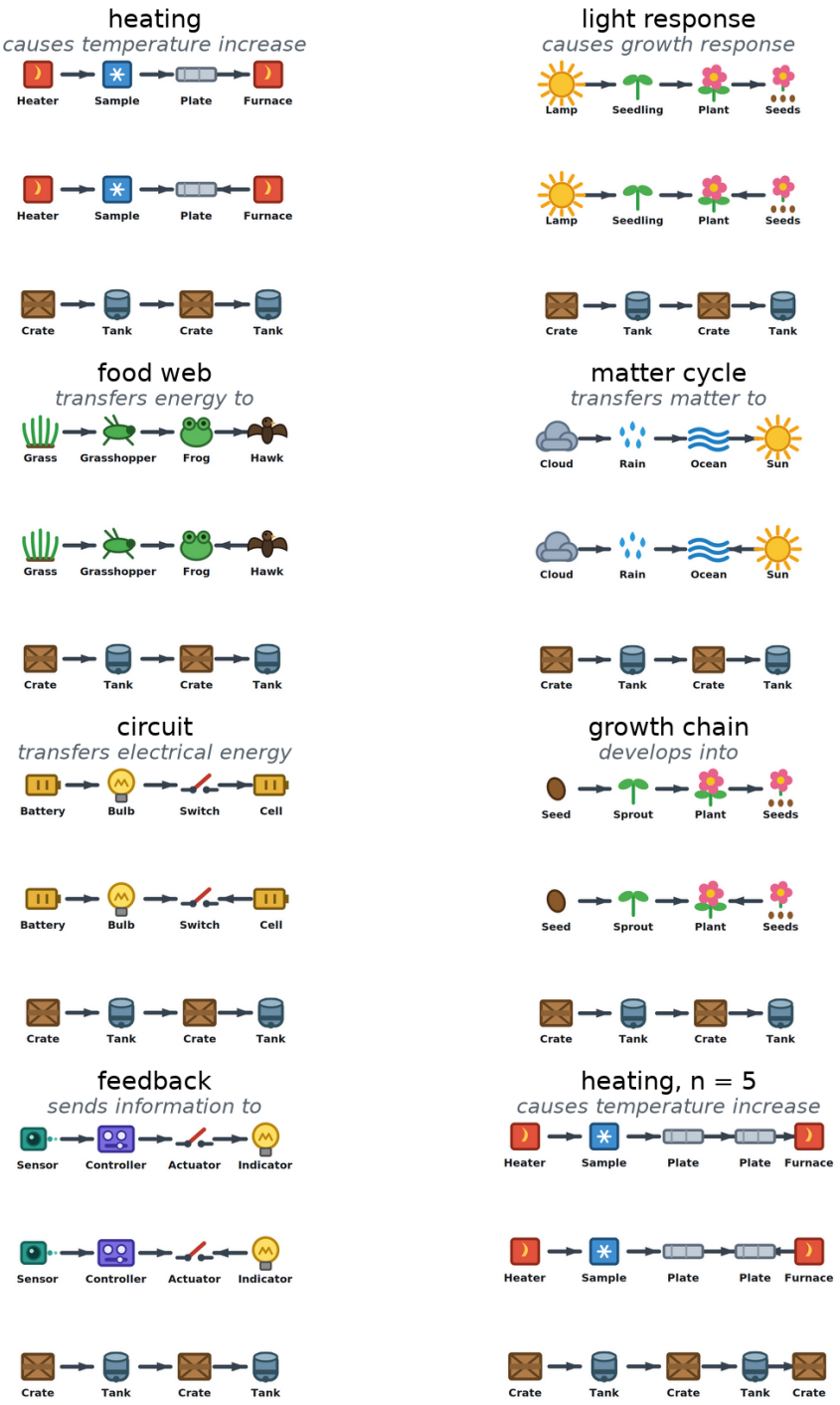

Figure 3: One base diagram from each scientific system. The top lane continues farther past the target arrow ($M \rightarrow X$), the middle one less far ($M \leftarrow X$), and the bottom lane is the crate–tank control lane. The two role lanes differ only in the direction of their last arrow. The last panel shows the five-node version of the heating system. The relation named under each system is not drawn in the image.

Control lane. The third lane has the same shape and alternates between a crate and a tank, entities that appear nowhere else in the corpus. We chose them after an earlier version reused entities across systems and the probe learned to recognize entity pairs instead (Appendix A.6).

Sixteen variants. Each system is built sixteen times by crossing four yes-or-no choices, each of which changes the graph rather than its look:

- **Which lane reaches farther:** lane A or lane B.
- **Lane length:** four or five nodes. A five-node lane repeats the next-hop entity ($S \to T \to M \to M \to X$), putting the remote arrow three steps away instead of two. Both roles appear at both lengths, so chain length cannot give away role.
- **Tail icon:** the system's tail entity, or a second copy of the next-hop entity.
- **Control direction:** the control lane's arrows point forward or backward.

Layouts, seeds and prompts. The three lanes are stacked in three rows, drawn in all six orderings. Every lane sits in every row exactly twice, so a lane's height on the page says nothing about its role. The same graph and layout always produce the same image; what varies is the noise the model adds before reading it, and we use two seeds per image, both kept in the same cross-validation fold. Each system has one fixed text prompt, shared by original and edited images, that names the kinds of objects but never says which way anything points.

A.2 From Base Diagrams to Model Passes

A base diagram is the unit for confidence intervals: every layout, seed and edit of it is another look at the same item. In computation it is many passes.

What the probe reads. The model sees the image as a 64×64 grid of tokens, each covering 16×16 pixels. For one target arrow of the heating system the readout region is 122 tokens: 84 covering the arrow itself and a 38-token border around them. It never overlaps the two remote arrowheads that decide the role, which are 9 and 6 tokens. The attention intervention of Appendix A.8 takes both lanes' target arrows as its queries, 156 tokens (84 and 72, without the border), because it is applied once for the whole layout rather than once per edit.

Variants, layouts and edits. Figure 4 shows all sixteen diagrams of the heating system; the other six follow the same pattern, giving $7 \times 16 = 112$. Figure 5 shows the six row orderings of one diagram, and Figure 6 the three edits made to each layout. Deletion and reversal are run separately.

The blocks we record. The four depth-spread blocks of Section 3 are 9, 18, 32 and 47 in Qwen-Image and 6, 11, 20 and 30 in Stable Diffusion 3.5; the four added between the last two are 35, 38, 41 and 44, and 22, 24, 26 and 28.

Passes. One experiment — a *cell* — fixes the model, the kind of edit and the noise level. Per layout and seed the model runs once on the original and once on each of the three edited images, so one diagram costs $6 \times 2 \times (1 + 3) = 48$ passes and a full 112-diagram cell costs 5,376 (Table 3). Across the project we ran 64 extraction runs and 220,718 passes (Table 4), counted from the rows each run saved rather than from the plan. A run is one extraction job; several runs may share a cell, as the redrawn versions do. That total includes the earlier stimulus designs that turned out to be flawed (Appendix A.6) and two runs of a follow-up we do not report; it excludes a few exploratory probes saved in an older format, the separate runs used to measure attention, and image generation.

Table 3: Passes through the model in one experiment. The redrawn-version row counts one redrawn version of the remote arrow, over 56 four-node diagrams.

Experiment	Diagrams	Layouts	Seeds	Edits	Original passes	Edited passes	Total
Main corpus	112	6	2	3	1,344	4,032	5,376
Redrawn version (each)	56	6	2	3	672	2,016	2,688
Attention cut, full or partial	56	3	2	3	336	1,008	1,344
Loop-versus-skip family	56	6	2	3	672	2,016	2,688
AI2D-RST textbook diagrams	349	1	2	2	698	1,396	2,094
Pairwise endpoint readout	112	6	2	4	1,344	5,376	6,720



Figure 4: The sixteen base diagrams of the heating system. Each title gives the variant number, the lane that reaches farther (A or B), the lane length n, the tail icon (`ext`: tail entity; `mid`: a second next-hop entity), and the control-lane direction (`fwd` or `rev`).

Table 4: Passes through the model, grouped by experiment. A *run* is one extraction job, so each group's passes are its runs times the per-run cost in Table 3: the redrawing group is $23 \times 2{,}688$, of which Table 8 reports twelve versions, and the last group is $3 \times 6{,}720$ pairwise plus one AI2D-RST and one loop-versus-skip run. Of the total, 54,346 are passes on original images and 166,372 on edited images. Not counted: the two-arrow, shifted, bent and branching families, the same-site readouts and the lane-separated cuts, so the total is a lower bound.

Group	Runs	Passes
Earlier stimulus designs (both models)	7	46,592
Main grid: 2 models $\times$ 2 edits $\times$ 2 noise levels, plus extra blocks	12	64,512
Redrawing the remote arrow	23	61,824
Attention cuts, partial cuts and one follow-up analysis	17	22,848
Pairwise readouts, AI2D-RST and the loop-versus-skip family	5	24,942
Total	64	220,718

A.3 Compute Resources and Runtime

All experiments ran on a shared server with eight NVIDIA L40S GPUs (46 GB each), two AMD EPYC 7763 CPUs and 1 TB of memory, using PyTorch 2.13.0 (CUDA 12.6 build), Transformers

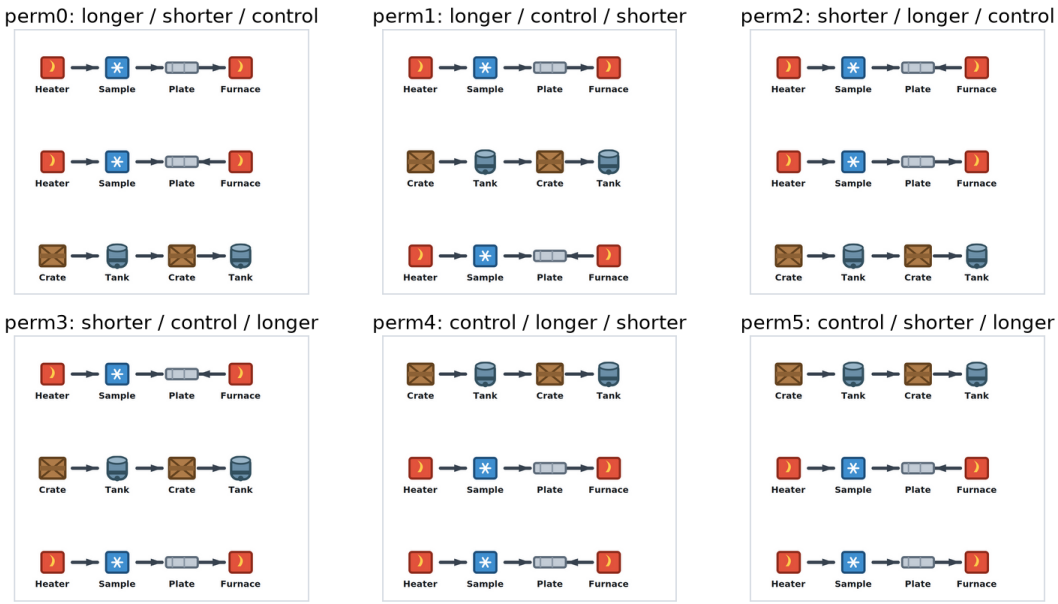

Figure 5: The six layouts of `ld_heating_00`. Each title lists the lanes from top to bottom: the lane that continues farther, the one that continues less far, and the control lane. Positions within a lane do not change.

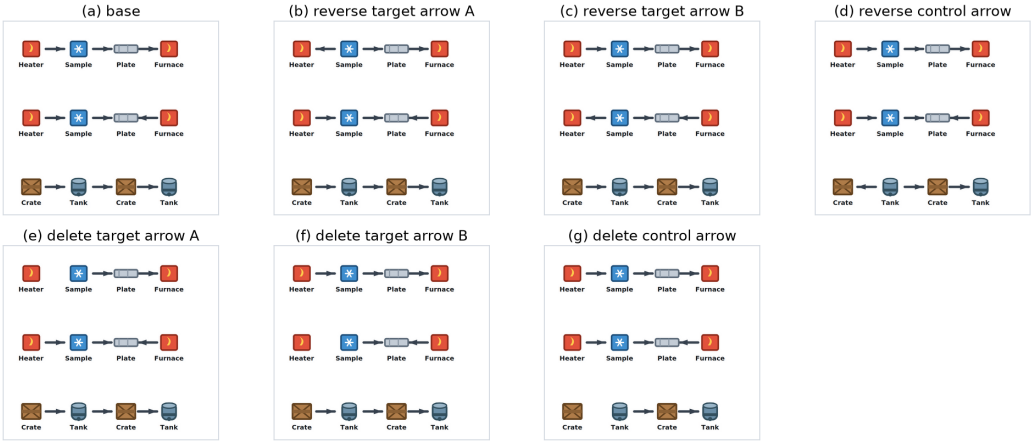

Figure 6: The six edits for one layout. (a) Original. (b)–(d) Reversing target arrow A, target arrow B and the control arrow. (e)–(g) Deleting the same three. One cell fixes the kind of edit, so a reversal cell uses (b)–(d) and a deletion cell (e)–(g); that is the three edited images per layout and seed in the arithmetic of Table 3. Every edited image is drawn from the edited graph, not modified pixel by pixel.

5.14.1, scikit-learn 1.9.0, NumPy 2.5.1 and a development build of diffusers 0.40.0 from source at commit `25d455f`; the pinned environment will be released with the code. Both models run in bfloat16. The Qwen-Image checkpoint (54 GB on disk) does not fit on one card and runs with model-level CPU offloading; Stable Diffusion 3.5 Large fits without it.

Each extraction cell is split into shards, one GPU each. We measured throughput from the start and end times of 179 shard logs, model loading and image encoding included (Table 5). A full 112-diagram cell costs about 2 GPU-hours on Qwen-Image and 1.3 on Stable Diffusion 3.5. A typical shard finished in 6 minutes on Qwen-Image and 4 on Stable Diffusion 3.5; the longest took 40. Applying these rates to the counted passes gives about 72 GPU-hours of extraction.

Table 5: Extraction compute by model. Throughput is the median over logged shards (interquartile range in brackets) and includes model loading; GPU-hours are the counted passes divided by that rate, so they are estimates.

Model	Forward passes	Passes per GPU-hour	GPU-hours
Qwen-Image	159,790	2,772 [2,686–2,972], 145 shards	≈ 58
Stable Diffusion 3.5 Large	60,928	4,271 [3,973–4,708], 34 shards	≈ 14
Total	220,718		≈ 72

Probe fitting, bootstrap intervals and label shuffles run on CPU: about a minute for a typical cell, 25 minutes for the longest, a sweep over five redrawn versions and eight blocks. Saved pooled activations take 32 GB and all saved outputs 35 GB. These figures exclude the separate runs used to measure attention, image generation, smoke tests, restarted runs and development time. Throughput was measured on the later cells, the ones with logged start and end times; earlier cells may have run at different speeds.

Part II The Probe and Its Checks

How an estimate is computed and checked, the two standard probing baselines run beside it, and the earlier signals that had simpler causes. Appendices A.4–A.6.

A.4 Checks Applied to Every Estimate

Section 3.5 asks three questions before a role estimate is read. This section gives the probe they are asked of, the exact form of each check, and where its thresholds came from.

The probe. Every role probe in the paper runs one pipeline, fixed in the first plan entry: the features are standardized, reduced to eight principal components fitted on the training rows only, and passed to an L_2-regularized logistic regression with $C = 0.003$. The position-only probe runs the same pipeline on the position features of Section 3; the full probe adds the readout of Equation 1. Four folds hold out whole base diagrams, so every layout and both noise seeds of a held-out diagram are scored by a probe that saw none of them; its twin, which draws the same images, can be in training (Appendix A.20). The nonlinear check replaces the logistic regression with a perceptron of two hidden layers (64 and 32 units). It needs its own penalty: the linear probe's, carried over, drives both probes to the majority class and reports a null from a dead instrument, so we take the setting with the largest validity score among those whose shuffled-label score straddles zero. This is the straddle test that Section 3.5 retires for judging estimates; here it only chooses among penalties, and every perceptron estimate we report is then judged by the magnitude rule.

Does the measurement work? The validity check compares target-arrow edits with control-arrow edits on the same features, folds and probe. It passes only if the lower end of its 90% interval lies above $\delta = 0.10$ bits *and* its accuracy beats the majority rate. The accuracy condition was added after bits alone let a probe through that could not discriminate: on an early set of reversal edits it gained $+0.128$ bits at accuracy 0.667, exactly the majority rate, by matching the base rate better than the position-only probe did.

Is zero really zero? The magnitude rule of Section 3.5 was calibrated on the 364 role estimates saved before it was written. The 219 of them then judged clean had offsets of at most 0.0098 bits, and among estimates above 0.10 bits the offset was a median 0.005 and a 90th percentile 0.096 of the estimate. Some analyses were specified with the older test of whether the shuffled interval straddles zero; every verdict we report is rescored under the magnitude rule from the stored offsets, and where the two tests disagree the table says so. Appendix A.18 gives the case that retired the older test.

Is a small result small enough? The margin $\delta = 0.10$ bits is the one fixed in the first plan entry and is used unchanged everywhere: an interval inside it is negligible, one that crosses it inconclusive.

Appendix A.19 checks the complementary failure, an estimate inside the margin produced by a probe that separates the classes but states its confidence too cautiously.

A.5 Standard Probing Baselines

Section 4.2 asks what standard probing practice would have concluded about each apparent signal. Beside the validity check and the label shuffle we ran the two controls the probing literature uses most for this, on the saved features of each case and with the probe, features and folds that produced it. Each case first had to reproduce its original estimate to within 0.01 bits (Table 6, first column). The decision rules were specified before any baseline was computed, with one prediction: that the remote-arrow signal would pass both (Appendix A.17).

Control-task selectivity. Following Hewitt & Liang (2019), each example's *type* is the pair of icons at the ends of the edited arrow, and every type is given a class at random from the task's own classes. The probe is trained on these control labels exactly as on the real ones. Selectivity is task accuracy minus control accuracy on held-out rows, averaged over five control seeds, with a 90% bootstrap interval over base diagrams (over scientific systems for endpoint identity, whose folds hold out layouts). A selectivity whose whole interval lies below 0.10 is a concern, and one whose whole interval lies above 0.10 is clean.

Online-code description length. Following Voita & Titov (2020), base diagrams are put in a random order and the labels are transmitted in blocks covering 0.1% to 100% of the data. Each block is coded by the same probe trained on everything before it, the first with a uniform code; for endpoint identity the order runs over layouts within each scientific system, in four blocks. We report codelength in bits per row for the real and the control labels, over five orderings. MDL selectivity is control minus task codelength. It is clean only if it exceeds 0.05 bits per row in every ordering, and a concern if its mean is at most 0.05 or its sign changes across orderings.

Table 6: Standard probing baselines on the four apparent signals. "Repro." is the reproduced ΔS (about which edge was edited for endpoint identity, about role otherwise). Accuracies are held-out means over base diagrams (over scientific systems for endpoint identity); codelengths are means over five orderings, in bits per row. Verdicts follow the rules specified in advance (selectivity / MDL). †The per-diagram estimator that produced this case cannot fit the control task, because the control label is constant within a base diagram; the values shown come from a pooled refit.

Case	Repro.	Task acc.	Control acc.	Selectivity [CI_{90}]	Task bits	Control bits	MDL sel.	Verdicts
Endpoint identity	$+0.539$	0.615	0.691	-0.076 $[-0.128, -0.021]$	1.633	1.022	-0.611	concern / concern
Chain length	$+0.553$	0.892	0.889	$+0.003$ $[-0.014, +0.020]$	0.554	0.513	-0.041	concern / concern
Displaced estimator†	-0.186	0.516	0.904	-0.388 $[-0.405, -0.370]$	0.999	0.332	-0.667	concern / concern
Remote arrow	$+0.935$	0.992	0.648	$+0.343$ $[+0.316, +0.372]$	0.279	1.065	$+0.786$	clean / clean

Both baselines flagged the two stimulus shortcuts: on the endpoint-identity design the probe learned the random type labels better than the real ones, and on the chain-length design about as well. Neither result names the objects or the chain length as the cause. On the remote-arrow signal both gave a clean result, as predicted, although its information comes from outside the readout region and depends on how the remote arrow is drawn (Appendix A.7). The displaced estimator cannot be tested with its own estimator, and its pooled refit is flagged mainly because the real labels carry no signal there, not because the refit detects the displaced zero.

One property of the designs matters for reading these numbers. The two target arrows of a diagram share their icons, so a type is nearly constant within a scientific system, and a probe can learn the control labels by recognizing the system. That is the control task exactly as defined, applied without modification, and it is why control accuracy is high wherever the real task is weak. It weakens the two flags, not the clean result on the remote arrow.

A.6 Four Apparent Signals and Their Causes

Four times in this project the probe seemed to have found relational structure and was reading something else. Each needed a different fix (Table 7), and Figure 7 shows the changes to the diagrams

that removed the first two. The fourth passed every standard control; Appendices A.7 and A.8 trace it.

Table 7: Four results that looked relational and were not. Values are ΔS in bits. The chain-length value is the original run; Tables 1 and 6 use its reproduction, $+0.553$.

What we saw	What the probe was using	How we found out	Before → after
The probe could tell which edge was deleted	Which objects sat at the ends of the arrow	Kept only the two edges drawn with identical icons and labels	$+0.539 \rightarrow -0.001$
A nonlinear probe found role where a linear one did not	How long the chain was	Redesigned the diagrams so both roles appear in short and long chains	$+0.556 \rightarrow +0.00002$
A precise, clearly nonzero estimate	A displaced estimator, whose "no information" point was not zero	Shuffled the labels: the estimate stayed near -0.099 instead of 0	-0.186 (no valid zero)
A strong role signal that passed every standard control	How the remote arrow is drawn	Redrew that arrow with the graph fixed, cut attention to its head, then split the role across two arrows	Redrawn: $+0.935 \rightarrow$ at most $+0.094$; cut: $+0.730 \rightarrow +0.005$; follows one arrow

Matching the two target arrows. Nothing near the target arrow may tell the two role lanes apart, so every diagram is checked on its rendered pixels, in every layout, before any activation is read. The two target arrows' drawn length must agree within 5%, their orientation within $5°$, their ink mass within a fixed radius of the midpoint within 5%, and the two aligned neighborhoods within 2% pixel by pixel. The radius stops short of the first node at which the lanes diverge, so the difference that sets the role lies outside the matched region. A diagram that fails any check is an error, not a warning. All 672 diagram–layout checks pass; the largest differences are 0.11% in ink and 0.40% pixel by pixel, with length and angle identical.

The objects at the ends of the arrow. Our first design asked a probe which edge of a diagram had been deleted. It succeeded clearly in Qwen-Image, $+0.539$ bits (CI_{90} $[+0.229, +0.837]$) at block 32 on layouts it had not been fitted on, beating a probe on the raw pixel difference by $+0.568$ bits. It was recognizing which objects each edge connected rather than where the edge sat in the graph. On the one pair of edges in that design drawn with identical icons and labels, two Battery→Bulb edges either side of a switch, the estimate fell to -0.001 ($[-0.303, +0.302]$, 15 of 32 diagrams correct). That interval is wide, but it excludes an effect the size of the original. The label shuffle sat at -0.216 bits, which against $+0.539$ leaves the estimate unreadable under the rule of Section 3.5, so the shuffle did flag this design, without saying why; the design had no control edge, so the validity check did not apply. A related leak appears in Figure 7a–b: because one probe is fitted on all systems together, an entity pair that is a target in one system and a control in another is tied to the wrong class. The final corpus uses identical icons and labels within each diagram and the same crate–tank control lane everywhere.

The length of the chain. Once the objects were matched, the perceptron found a strong signal in Stable Diffusion 3.5 that a linear probe on the same rows did not: $+0.556$ against $+0.006$ bits at block 20, $s = 0.4$ ($+0.553$ when reproduced for Table 6). The lane that continued farther was also the longer one (Figure 7c), and a chain's length is easy to pick up even though nothing near the arrow shows it. The validity check passed and the label shuffle was clean (-0.0007 bits; the straddle test in use then flagged it only because its interval was tight), so neither check pointed to the cause. On the redesigned diagrams, where both roles appear in short and long chains (Figure 7d), the same probe at the same block reads $+0.00002$.

A displaced estimator. An early analysis fitted a separate probe to each base diagram. It returned a precise, clearly negative role estimate, -0.186 bits ($[-0.204, -0.168]$), and with shuffled labels, which should return zero, it returned -0.099. That is a bias in how the estimate is computed, not noise, so more data would not remove it. One probe fitted across all training diagrams, each diagram

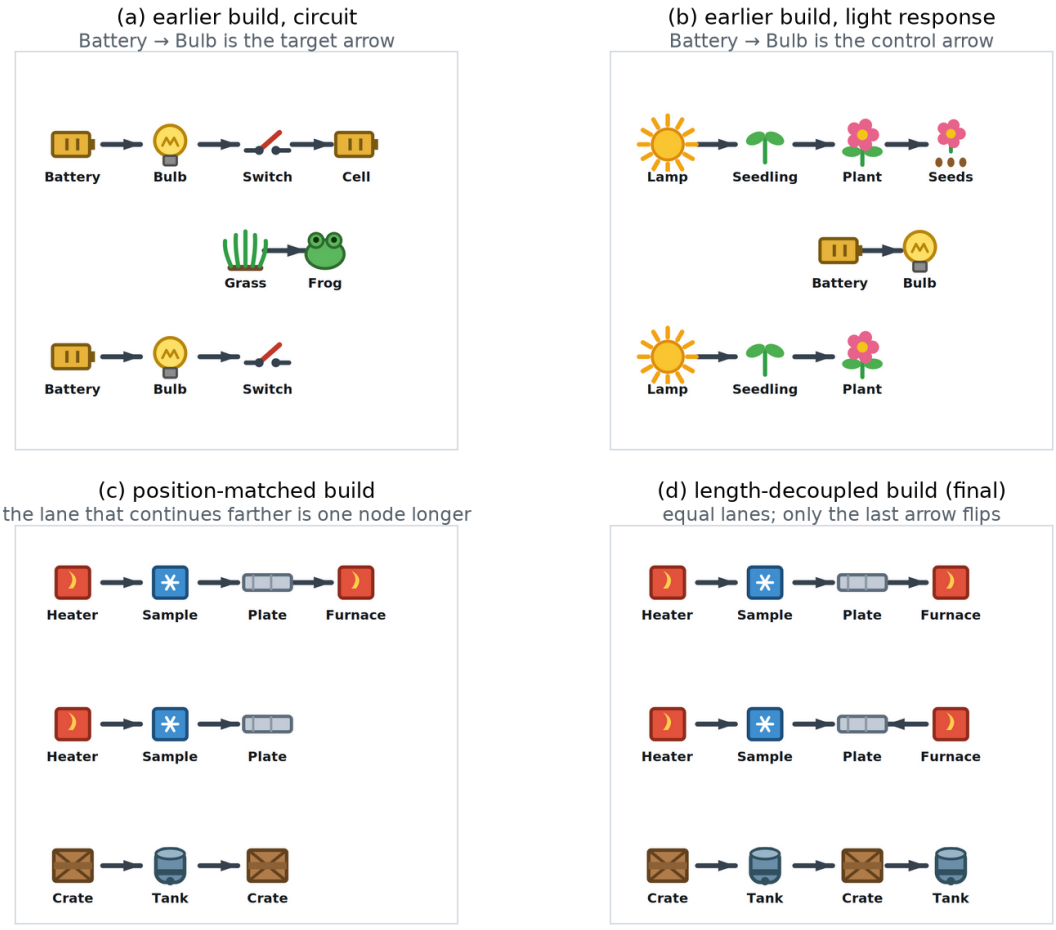

Figure 7: How the diagrams changed. (a)–(b) In an earlier version each scientific system drew its control lane with its own entities, and some were another system's target pair: Battery → Bulb is the target arrow in the circuit system but the control arrow in the light-response system. (c) In the next version, the lane that continued farther was also one node longer. (d) The final version: both role lanes have the same length and icons and differ only in their last arrow, and every control lane uses crates and tanks.

held out in turn, brings the shuffle on the same 96 diagrams back to $+0.001$, and every estimate we report uses it.

Part III Locating the Source

The two experiments behind Section 3.6: redrawing the remote arrow, and cutting the readout region's access to it. Appendices A.7–A.8.

A.7 Redrawing the Remote Arrow

Distances in this appendix are measured from the target arrow's bounding box to the remote arrow, not from the center of the readout region to the arrowhead centroid as in Section 3; the same geometry is about 159 px longer under that convention, so the standard drawing at 263 px here is the 422–459 px reported there.

Distance does not matter; drawing does (Table 8). Moving the arrow farther away leaves the signal intact — the farthest version is among the strongest — while small changes to how it is drawn remove most of the signal, including when the arrowhead stays fully visible. Sliding the neighboring

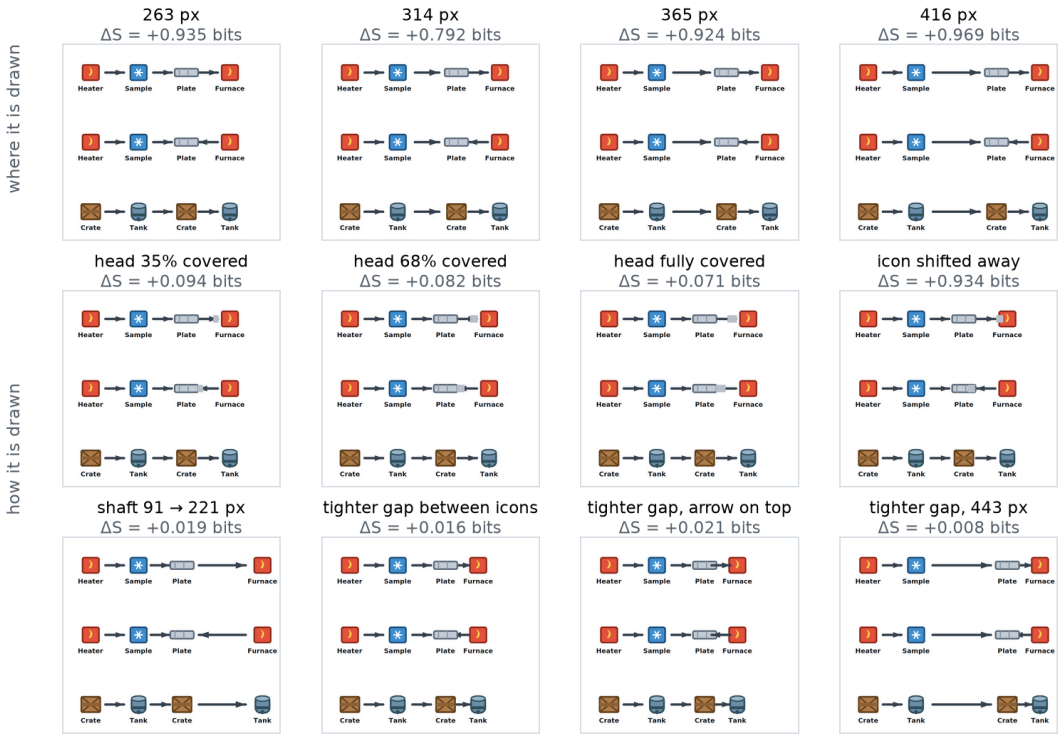

Figure 8: Redrawing the remote arrow in `ld_heating_00`. Top row: the arrow moved to 263, 314, 365 and 416 pixels from the target arrow. Middle row: the neighboring icon slides over the arrowhead until it is 35%, 68% and fully covered, and the same shift in the opposite direction, which leaves the arrowhead visible. Bottom row: a stretched shaft, a tighter gap between the icons, the same tighter gap with the arrow drawn on top, and the tighter gap at 443 pixels. All twelve versions of Table 8 appear here. Distances are measured from the target arrow, as stated at the head of this section. Coverage is measured on the drawn image. Panel titles give the result from Table 8.

icon the same distance *away* from the arrowhead costs nothing, so the drop is not caused by moving the icon.

Table 8: Role signal (ΔS, bits) when the remote arrow is redrawn. Qwen-Image block 38, reversal, $s = 0.2$, 56 four-node diagrams, six layouts. Accuracy is against a position-only 0.500 throughout. The equivalence margin is ± 0.10 bits. Every label-shuffle offset is within 0.009 bits of zero, clean under the rule of Section 3.5; the straddle test in use when these ran flagged several of the low values. The standard row comes from this sweep's own extraction, which reproduces the four-node estimate of Section 4.1 ($+0.935$ at accuracy 0.992 there, 0.993 here).

	Version of the remote arrow	ΔS	Acc.
Where it is drawn	263 pixels away (standard)	$+0.935$	0.993
	314 pixels away	$+0.792$	0.952
	365 pixels away	$+0.924$	0.985
	416 pixels away	$+0.969$	1.000
How it is drawn	arrowhead 35% covered	$+0.094$	0.590
	arrowhead 68% covered	$+0.082$	0.608
	arrowhead fully covered	$+0.071$	0.612
	icon shifted away, arrowhead visible	$+0.934$	0.995
	shaft stretched from 91 to 221 pixels	$+0.019$	0.547
	tighter gap between icons	$+0.016$	0.561
	tighter gap, arrow drawn on top	$+0.021$	0.568
	tighter gap, 443 pixels away	$+0.008$	0.548

Because the graph, the labels and the target arrow are identical throughout, this rules out a code for role that holds however the remote arrow is drawn. It does not isolate one property of the drawing: covering the head, stretching the shaft and crowding the icons each remove the signal, and drawing the crowded arrow on top of the icons, which uncovers its head, does not bring it back.

The arrow stays legible in that last version. Read at its own region rather than at the target arrow, a probe recovers its drawn direction at every block we recorded ($+0.93$ bits at block 38, accuracy 1.000, against a position-only baseline of exactly 0.500). Under the standard rendering the same probe reads $+0.96$ bits at accuracy 1.000 at block 38. So redrawing does not change whether the direction is encoded where the arrow is drawn; it changes whether it reaches the readout region.

The same crowding explains the five-node diagrams, where the signal is also absent. Their remote arrow sits between icons as closely spaced as in the tighter-gap version, and the two agree ($+0.017$ and $+0.016$ at block 38), so the five-node null does not separate hop count from drawing.

These comparisons point at the drawing but cannot prove it, since two separately drawn images may differ in ways we did not intend. That needs a change to the model rather than the picture.

A.8 Blocking Access to the Remote Arrow

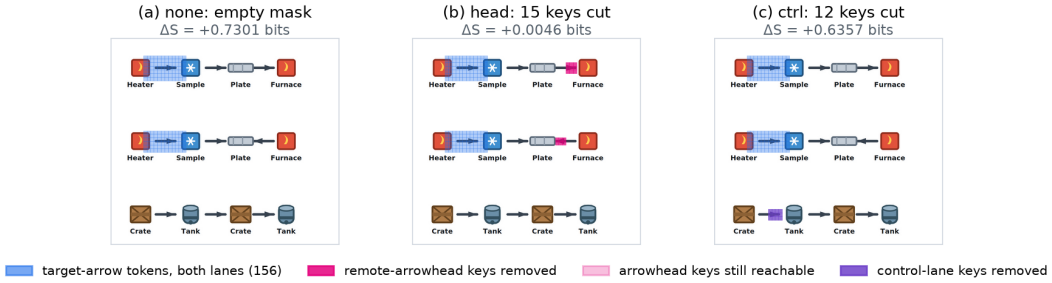

Figure 9: Which tokens are blocked. The queries are the tokens of both lanes' target arrows, 156 in this diagram (84 and 72): the arrow part of each edit's readout region, without its 38-token border. (a) Nothing blocked; (b) the queries cannot attend to the two remote arrowheads (15 tokens); (c) they cannot attend to the control arrowhead instead (12 tokens). Panel titles give the result from Table 9.

Blocking the remote arrowheads removes almost the whole signal; blocking the control arrowhead leaves most of it (Table 9). The probe still tells target-arrow edits from control-arrow edits in all three conditions (accuracy 0.951–0.980), so blocking did not break the measurement. Blocking only part of the arrowhead gives a steep decline: at block 38 a quarter of the tokens leaves $+0.156$ bits, a half $+0.030$ and three quarters $+0.010$, so a quarter already removes about four fifths of the signal. The fractions tested are 0, 0.25, 0.5, 0.75 and 1, rounded to whole tokens (4, 8 and 12 of 15), so where inside the first quarter the fall happens is unresolved.

Table 9: Role signal (ΔS, bits) when attention to the remote arrowhead is blocked. Three layouts (`perm0`, `perm2`, `perm4`), 56 four-node diagrams; Qwen-Image block 38 unless noted. Layout count moves the position-only baseline, so compare these rows with each other and not with Table 8. Role accuracy is in brackets, against a position-only 0.464 on these three layouts.

	Nothing blocked	Remote arrowhead	Control arrowhead
Qwen-Image, block 38	$+0.7301$ (0.938)	$+0.0046$ (0.525)	$+0.6357$ (0.866)
Qwen-Image, block 47	$+0.9432$ (0.999)	-0.0048 (0.476)	$+0.9459$ (1.000)
Stable Diffusion 3.5, block 30	$+0.0212$ (0.571)	-0.0005 (0.501)	$+0.0168$ (0.552)

Separating the lanes. Two further cuts, added after the plan, split the queries by lane. In the *own* cut each lane's target arrow cannot attend to its own lane's remote arrowhead; in the *cross* cut, to the other lane's. The queries are the joint cut's, split by lane, so each target loses 6 to 9 key tokens where the joint cut removes 15 and the control cut 12, and no query lies in a blocked region. The

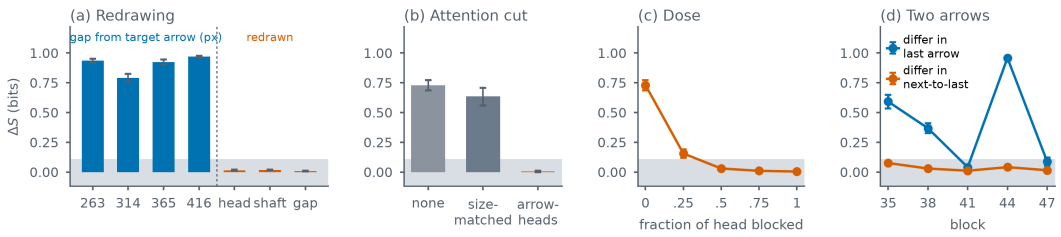

Figure 10: **Tracing the signal to its source** (Qwen-Image, reversal edits, $s = 0.2$, four-node diagrams; bars and points are ΔS with 90% intervals, shaded band the ± 0.10-bit margin). **(a)** Redrawing the remote arrow, block 38. **(b)** Cutting the readout region's attention to the arrowheads, block 38 on three layouts, where the unmasked reference is $+0.730$ rather than $+0.935$. **(c)** Blocking a fraction of the arrowhead's tokens. **(d)** The two-arrow family, by block.

same code with both arrowheads blocked for both lanes reproduces the joint cut bit for bit. The own cut matches the joint cut at every late block, within 0.008 bits when paired by diagram, and the cross cut raises the signal at every late block, every paired interval over twin pairs lying above zero (Table 10). Validity passes under every cut at blocks 35 to 47 (lowest interval end 0.354, accuracy 0.90 to 0.98), and every late-block shuffle is clean. Two accounts of the rise fit and neither was tested: the other lane's arrowhead, which always points the other way, interferes with the readout, or removing it shifts the renormalized attention toward the own arrowhead. The own cut shows that access to the lane's own arrowhead is necessary, not that it is the only source.

Table 10: Attention cuts separated by lane. Qwen-Image, reversal, $s = 0.2$, the three layouts and 56 four-node diagrams of Table 9. ΔS in bits, role accuracy in brackets, against a position-only 0.464. "Both heads" is the joint cut of Table 9. Every shuffle is clean. Run after the plan.

Block	Nothing blocked	Own lane's head	Other lane's head	Both heads	Control arrowhead
35	$+0.614$ (0.909)	-0.004 (0.490)	$+0.837$ (0.973)	$+0.003$ (0.540)	$+0.621$ (0.874)
38	$+0.730$ (0.938)	-0.003 (0.496)	$+0.909$ (0.985)	$+0.005$ (0.525)	$+0.636$ (0.866)
41	$+0.174$ (0.711)	$+0.001$ (0.518)	$+0.901$ (0.985)	$+0.003$ (0.527)	$+0.146$ (0.680)
44	$+0.485$ (0.856)	-0.004 (0.501)	$+0.948$ (1.000)	-0.002 (0.503)	$+0.436$ (0.820)
47	$+0.943$ (0.999)	-0.003 (0.499)	$+0.966$ (1.000)	-0.005 (0.476)	$+0.946$ (1.000)

Figure 10 collects this section's numbers with those of Appendix A.7 and the two-arrow family of Appendix A.9.

Access is not attention weight. We chose which tokens to block by where the arrowhead is, never by how much attention they receive, and the partial blocks were drawn at random. These experiments therefore ask whether access to a region matters, not whether attention weights rank tokens by importance. The two come apart: at blocks 35 and 38 the shaft-stretched version directs about as much attention to its arrowhead as the versions with the strongest signal, while carrying almost no role information. Blocking also renormalizes the remaining attention, so this is not a clean deletion of one computation; the empty mask, the control mask and a version with no signal to lose all limit that concern without removing it.

Part IV Further Diagram Families

The five families of Section 3.6, each holding the target arrow fixed and changing either what decides the role or where it is drawn. Appendices A.9–A.13.

Figure 11 shows one diagram from the base corpus and one from each of the five families. Every panel is rendered from the same code that produced the stimuli, so it is the image the model was given, blank rows between lanes removed.

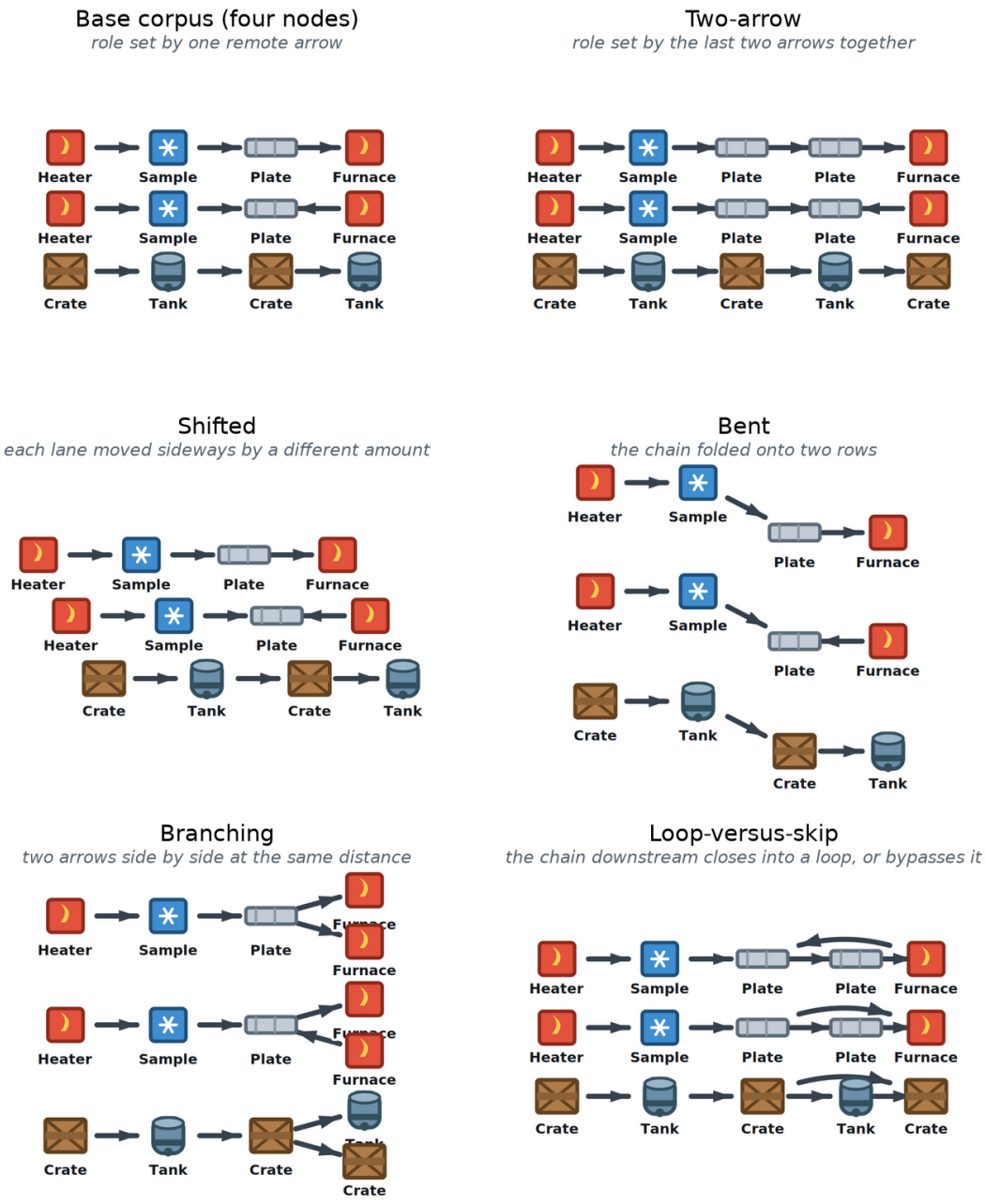

Figure 11: One example from the base corpus and from each further family, all from the heating system. **Base corpus:** the role of the target arrow (Heater→Sample) is set by the last arrow alone, which points onward in the top lane and back in the middle one. **Two-arrow:** five nodes, and the chain continues only if the last two arrows both point forward. **Shifted:** each lane is moved sideways by its own amount; the geometry inside a lane is pixel-identical to the base corpus. **Bent:** the chain is folded onto two sub-rows. **Branching:** the chain splits into two mirror-image tips at equal distance, and continues only if both point outward. **Loop-versus-skip:** a curved arrow over the middle of the chain either closes a loop or skips a node. The bottom lane of every panel is the control lane.

A.9 Two-Arrow Family

The last arrow sits 435 pixels from the target arrow, farther than any version in Table 8 that is not also crowded. So that a null could not be blamed on distance, a control ran in the same session with

the same seeds and noise: four-node diagrams whose single remote arrow is moved to 437 pixels at the same icon spacing. It is strongly positive at every late block, which makes the comparison readable.

Table 11: Role signal (ΔS, bits) when the role is decided by two arrows. Qwen-Image, reversal, $s = 0.2$, 56 diagrams, six layouts, 28 base diagrams per subset. "Last arrow" holds the next-to-last arrow fixed between the two lanes, so only the last one separates them; "next-to-last" is the mirror. The control is a four-node diagram with its single remote arrow at the same distance. Role accuracy is in brackets, against a position-only 0.500. The equivalence margin is ± 0.10 bits. The two "twins" columns refit both subsets with each pixel-identical twin pair in one fold (Appendix A.20), after the plan. * label-shuffle flag.

Block	Last arrow	Next-to-last arrow	Last, twins	Next-to-last, twins	Distance control
35	$+0.591^*$ (0.914)	$+0.076$ (0.624)	$+0.552^*$ (0.866)	$+0.091$ (0.658)	$+0.970$ (1.000)
38	$+0.368$ (0.796)	$+0.030$ (0.576)	$+0.525$ (0.860)	$+0.026$ (0.592)	$+0.968$ (1.000)
41	$+0.040$ (0.601)	$+0.011$ (0.555)	$+0.046$ (0.583)	$+0.011$ (0.562)	$+0.932$ (0.995)
44	$+0.959$ (1.000)	$+0.041$ (0.613)	$+0.956$ (1.000)	$+0.049$ (0.613)	$+0.986$ (1.000)
47	$+0.087$ (0.624)	$+0.015$ (0.574)	$+0.292$ (0.705)	$+0.007$ (0.554)	$+0.982$ (1.000)

The probe separates the pairs that differ in the last arrow at blocks 35, 38 and 44, while the pairs that differ in the next-to-last arrow never clear the margin; with twins in one fold, block 35 reaches $+0.091$ $[+0.070, +0.111]$, which crosses it. The last-arrow subset is not monotone in depth: at block 41 it falls inside the margin, at $+0.040$, and at block 47 its $+0.087$ $[+0.056, +0.122]$ crosses the margin; with twins in one fold block 47 clears it, $+0.292$ $[+0.180, +0.403]$, while block 41 stays inside. The separation thus appears at three or four of the five late blocks rather than from some depth onward. That is what a readout of one arrow predicts, and a representation of the role would have to separate both, since either arrow alone leaves the role undetermined. Read with Appendix A.7, where redrawing that same last arrow removes the signal, this says the decoded variable is the drawn direction of one arrow rather than the role it helps define.

The arrow the probe reads is also the *farther* of the two, at 435 pixels against 231, and the nearer one never clears the margin. The distance control excludes distance as the explanation, and every arrow here is drawn at the same wide spacing, so crowding does not explain it either. What distinguishes the two is that one terminates the lane. Reading each arrow at its own region settles whether the earlier one is encoded at all: it is, at ceiling, so its absence at the readout region is a failure to arrive rather than an absent code (Appendix A.10). That same-site reading rests on accuracy, not bits, for the reason set out for the branching family in Appendix A.12: a within-diagram shuffle cannot permute a label that is constant inside the diagram. Our claim is accordingly the narrow one: the probe follows a single arrow rather than combining two.

Three limitations bound it. The conjunction of two directions is linearly separable, so this compares one arrow against two, not a linear code against an integrated one. The next-to-last arrow is below the margin, which is not the same as absent: its intervals exclude zero, but they also do so at blocks 9, 18 and 32, where the control is itself null, so the sign says nothing at this precision. Accuracy says more — that subset scores 0.56–0.62 at blocks 35–47 against 0.51–0.52 at blocks 9–32, a trace too small to clear the margin, but not zero. And at block 44 the probe pooled over both subsets reaches $+0.465$ bits at accuracy 0.833 ($+0.432$ at 0.814 with twins in one fold), above what any readout of one arrow can reach across the family. The last arrow points onward in three lanes of four (all 56 lanes that continue farther and the 28 shorter lanes of the next-to-last subset), and two thirds of those continue farther, so a probe that reads only that arrow has a best calibrated loss of $0.75\,H(1/3) = 0.689$ bits: a ceiling of 0.311 bits at accuracy 0.750, computed in the plan entry before extraction. The pooled estimate exceeds both, so the readout at block 44 is slightly better than one arrow alone, even though the next-to-last subset is not readable on its own. The pooled figure is not an average of the two subsets: it comes from a probe fitted on all 56 diagrams, while each subset probe is fitted on 28, and with twice the training data it can exceed the equal-weighted mean of the two subset accuracies, 0.807, which is why 0.833 sits above it. We read the experiment by the per-subset criterion, the contrast the family was built to make. The pooled block-44 figure is the caveat on that reading, and it does not exclude a small amount of integration at that block.

A.10 Direction at Each Arrow's Own Region

The probe at the target arrow separates pairs that differ in the last arrow and barely registers pairs that differ in the next-to-last one (Appendix A.9). That is consistent either with the model not encoding the earlier arrow's direction at all or with encoding it where the readout region cannot reach it. The two readings differ at one place, the arrow itself, so we moved the readout there: each arrow's own full edge region, pooled on the base image, asking whether its drawn direction is recoverable. The region is the whole edge rather than the arrowhead, for the reason given in Appendix A.18. Measured before extraction, each region's center sits at one position for both drawn directions (0.60 of the canvas width for the next-to-last arrow, 0.80 for the last), the two lanes' regions are identical in x in all 336 layout checks, and the two arrows' regions never overlap, so an estimate at one arrow cannot be reading the other.

Because the pair inside a diagram shares one arrow's direction, the 28 diagrams whose pair differs in the next-to-last arrow score that arrow's region and the other 28 score the last arrow's. The last arrow is the sensitivity check, in the same family, at the same spacing, with the same probe and folds.

Table 12: Direction of each arrow, read at that arrow's own region. Qwen-Image, $s = 0.2$, $n = 28$ base diagrams per site, position-only baseline accuracy 0.500 at every cell. * label-shuffle flag; † offset too large to read in bits (Section 3.5); the test specified in advance for this analysis asked only whether the shuffled interval straddles zero.

	Last arrow		Next-to-last arrow	
Block	ΔS	Acc.	ΔS	Acc.
9	$+1.038^{*}$	1.000	$+1.020^{*}$	1.000
18	$+1.024^{*}$	1.000	$+0.971^{*}$	0.996
32	$+1.125^{*}$	1.000	$+1.108^{*}$	1.000
35	$+1.167^{*}$	1.000	$+1.164^{\dagger}$	1.000
38	$+1.169^{*}$	1.000	$+1.163^{\dagger}$	1.000
41	$+1.160^{*}$	1.000	$+1.163^{\dagger}$	1.000
44	$+1.156^{*}$	1.000	$+1.154^{\dagger}$	1.000
47	$+1.165^{*}$	1.000	$+1.158^{\dagger}$	1.000

Both arrows are decodable at every block, at accuracy 1.000 (0.996 at one cell) against a position-only 0.500. In bits, the next-to-last arrow reads with a flag at blocks 9, 18 and 32 ($+1.020$, $+0.971$ and $+1.108$); at blocks 35–47, where the readout at the target arrow carries the signal being explained, its shuffle offset is too large to read the bits, and the claim there rests on accuracy alone. Read with Appendix A.9, this says the earlier arrow's direction exists where it is drawn and mostly does not reach the readout region.

Three limits go with the numbers. No cell has a clean label shuffle under the rule of Section 3.5. The test specified in advance for this analysis asked only whether the shuffled interval straddles zero, which five cells passed; rescored by magnitude, the last arrow's offsets run from 0.017 to 0.083 bits and the next-to-last's from 0.022 to 0.044 through block 32, all flagged, and reach 0.14 to 0.16 from block 35 on, which against estimates near $+1.16$ is too large to read. No cell was discarded or refitted at another probe rank. The estimates exceed one bit for a binary label because the position-only probe scores worse than chance on held-out diagrams, not because more than one bit is present, which is why the accuracy column is the honest summary. And decodability at an arrow is a local visual property, not a representation of role: it says where a drawn direction is available, not that the model uses it.

A.11 Shifted and Bent Layouts

These two were specified in advance together. In *shifted*, each lane is translated horizontally by its own amount while the geometry inside the lane stays pixel-identical. In *bent*, the chain occupies two sub-rows, with the distance from the readout region to the remote arrowhead held within 0.72 pixels of each item's twin in the original corpus. Shuffle verdicts follow the magnitude rule of Section 3.5,

rescored from stored offsets; the straddle test specified with them flags two cells that the magnitude rule reads as clean.

Table 13: Role signal (ΔS, bits) with probe accuracy, from a probe *refit* within each family (the frozen-probe transfer is in the paragraph below), on layouts the original probe was not fitted on. Position-only accuracy is 0.500 throughout and the validity check passes in every cell. Every shuffle offset is within 0.005 bits of zero.

	Shifted		Bent	
Block	ΔS	Acc.	ΔS	Acc.
35	$+0.702$	0.920	$+0.005$	0.534
38	$+0.859$	0.970	$+0.007$	0.548
44	$+0.564$	0.849	$+0.003$	0.530
47	$+0.524$	0.847	$+0.002$	0.522

With the probe frozen. A probe fitted on the original corpus and applied unchanged keeps most of its accuracy when lanes are shifted and loses all of it when the chain is bent: at block 38, 0.992 within the original corpus against 0.916 $[0.884, 0.944]$ shifted, and at block 47, 0.949 against 0.944 $[0.930, 0.957]$. On the bent family the same frozen probe reaches 0.560 $[0.544, 0.578]$ at block 38 and 0.589 at block 47, near the position-only 0.500.

The attention cut repeats on the shifted family. At block 38, with the image fixed, an empty mask leaves $+0.867$ bits at 0.981 accuracy; cutting the readout region's access to the remote arrowheads takes accuracy to 0.510, while a size-matched control cut leaves it at 0.984. The three cut cells carry shuffle offsets of about 0.011 bits, which against a cut estimate of $+0.006$ leaves that estimate unreadable in bits under the rule of Section 3.5, so the comparison rests on accuracy.

On the bent family the validity check still passes, 0.838 against its position-only 0.667, so the null is an absence rather than a dead instrument, and cutting attention there changes nothing, as expected when there is nothing to remove: all three mask conditions sit between -0.003 and -0.005 bits.

What the bend does not settle. Bending removes the signal, but it also enlarges the pooled readout region by about 5% and changes how that region lines up with the model's token grid: a vertical displacement that is not a whole number of patches changes how many tokens the readout averages over. This family cannot separate the displacement from that change. We report the null and do not attribute it.

A.12 Branching

Each diagram carries two equidistant branches and the target arrow's role is fixed by both directions together. Matched pairs vary one branch and hold the other, so a readout of the role must separate both kinds of pair. Because the two tips are mirror images at the same distance, and both branches end the chain at the same position in it, neither distance nor chain position singles one branch out, which removes the two explanations the two-arrow family could not separate.

At the readout region, nothing. Across blocks 35, 38, 44 and 47 and both subsets, accuracy runs from 0.490 to 0.521 against a position-only 0.500, all eight ΔS estimates lie between $+0.0001$ and $+0.0006$ bits with every interval inside the ± 0.10-bit margin, and the validity check passes in all eight cells (0.83–0.88 against its position-only 0.667). This is a negligible result on a working instrument. It was also not one of the two outcomes we specified in advance, which were that the probe would follow one branch or separate both; we record it as it came out.

Decodable at each arrow's own region. Because that null cannot distinguish an absent encoding from one that does not arrive, we added a same-site analysis afterwards, entered in the append-only plan file before it ran but after the branching result was known. Read at its own arrow, each branch direction is recoverable at accuracy 1.000 against a position-only 0.500 at all eight blocks recorded, both for a tight region and for one size-matched to the readout region.

Why that claim rests on accuracy. The bit estimates for the same-site analysis are inflated. Part of the corpus carries a label that is constant within a bootstrap cluster, and a within-item shuffle cannot permute a constant, so the shuffled control is not a null for those diagrams. Restricting to the diagrams whose label varies within a cluster removes the inflation and leaves accuracy unchanged, which is why we quote accuracy and leave the bits uninterpreted.

One drawing note. In these diagrams the upper tip's text label is drawn across the lower tip's icon (Figure 11). We checked whether this occludes a branch arrowhead, since occlusion is what removes the signal elsewhere in this paper: measured as the fraction of the arrowhead's inked area that survives in the finished image, the upper branch averages 0.939 and the lower 0.921, against 0.916 for the remote arrow of the base corpus. Both branches are therefore at least as exposed as the arrow that does produce a signal, and the small up-down difference would predict one subset surviving rather than both failing.

Branch direction is decodable where it is drawn and absent at the readout region. That is consistent with limited transmission and with limited combination, and this design separates neither.

A.13 Loop-versus-Skip Family

Everything above concerns one kind of role: whether a chain continues or stops. To see whether the picture changes for a different one, we built a family in which each lane is a five-node chain with one curved arrow over the middle (Figure 11). In one lane that arrow points backward and closes a loop; in the other it points forward and skips a node. The target arrow is drawn at exactly the same pixels as before. The two lanes have the same number of nodes and arrows, the same number of reachable nodes, the same longest path, and almost the same total drawn line (within 0.91%). The family has 56 diagrams, seven systems with eight variants each.

The probe finds no role signal here: estimates stay between -0.0011 and $+0.0020$ bits at every block checked, at chance accuracy, and the validity check passes throughout. Shuffle verdicts follow the magnitude rule of Section 3.5, rescored from the stored offsets, which run from -0.0064 to -0.0023 bits and are therefore clean at every block; the straddle test in use when these ran flagged three of them. One caveat keeps this from being a strong result: the validity check is weaker than in the main corpus (accuracy 0.81–0.88 against 0.95–1.00), so this family is a less sensitive test. It shows that the earlier signal does not carry over to this kind of role, but not why.

Part V Supporting Analyses

A pairwise endpoint readout, an exploratory transplant and real textbook diagrams: support for the main results, never the only basis for a claim. Appendices A.14–A.16.

A.14 Pairwise Endpoint Readout

Every readout in the main text averages the activation change over the edited arrow. That average could hide information carried by the arrow's two ends separately, and which end is which is exactly what makes a relation directed. An earlier version of this readout kept the ends apart but ran on deletion at one block, where there is no role signal to find, and chose its settings after seeing the data. We ran it once more, on the one signal that survives, with every choice specified in advance.

Design. We re-extracted the 112 base diagrams for Qwen-Image under reversal at $s = 0.2$, with all six layouts, both noise seeds and blocks 35, 38, 41, 44 and 47 read in one pass. Each layout has four edits: the two target arrows, the control arrow, and a *pair control*, the next arrow along the same lane, which differs from the first target arrow in which pair of objects it connects. The readout pools the tokens over the source node and over the target node separately, and the probe sees both vectors, their difference and their product, on a shared 32-dimensional basis fitted on training rows only. As a reference, the pooled readout, the mean over the edited arrow without the one-token border, is fitted on the same rows. The probe is linear with $C = 10^{-4}$, the value the earlier version's saved calibration had chosen, not tuned here. Decisions use the 56 four-node diagrams at block 38; the other blocks are descriptive.

Gates and stopping rule. Three checks come first, in order: the validity check; a pair-identity check (first target arrow against pair control), which must also beat a probe on the raw pixel difference with a 90% interval above zero; and the label shuffle, whose interval must include zero. The stopping rule allowed one extraction and one analysis, with reruns only for failures unrelated to results; no other readout, regularization, block or model in the cycle; both scripts frozen by checksum; and a reading fixed for every outcome. The run completed with no reruns. This validity gate asked only for the bit criterion, whereas the main check also requires accuracy above the majority rate. Block 38 meets both (0.694 against 0.667); at blocks 44 and 47 the pairwise probe sits at the majority rate (0.668 and 0.667), so those blocks would fail the main check too, besides the pixel criterion that already makes them unreadable.

Table 14: Pairwise readout on the surviving signal (Qwen-Image, reversal, $s = 0.2$, 56 four-node diagrams). "Validity" gives bits and accuracy against a majority rate of 0.667; "beats pixels" is the activation-minus-pixel ΔS on pair identity; role gives bits with a 90% interval, and accuracy against 0.500. Block 38 of the pairwise readout is the decision specified in advance. * shuffled interval excludes zero, the test specified for this analysis; under the magnitude rule of Section 3.5 every shuffle in the table is clean.

Readout	Block	Validity (acc.)	Beats pixels	Role ΔS [CI_{90}]	Role acc.	Shuffle	Outcome
Pairwise	35	$+0.188$ (0.714)	$+0.0175$	$+0.058$ $[+0.055, +0.062]$	0.961	-0.0002	negligible
	38	$+0.193$ (0.694)	$+0.0094$	$+0.045$ $[+0.042, +0.048]$	0.878	-0.0002	negligible
	41	$+0.190$ (0.706)	-0.0006	—	—	—	unreadable (pixel gate)
	44	$+0.147$ (0.668)	-0.0031	—	—	—	unreadable (pixel gate)
	47	$+0.140$ (0.667)	-0.0049	—	—	—	unreadable (pixel gate)
Pooled (reference)	35	$+0.691$ (0.992)	$+0.6310$	$+0.761$ $[+0.743, +0.778]$	1.000	-0.0078^{*}	shuffle flag
	38	$+0.722$ (0.999)	$+0.6260$	$+0.738$ $[+0.717, +0.757]$	1.000	-0.0066	role found
	41	$+0.703$ (1.000)	$+0.6247$	$+0.695$ $[+0.677, +0.712]$	1.000	-0.0017	role found
	44	$+0.679$ (0.996)	$+0.6420$	$+0.716$ $[+0.702, +0.730]$	1.000	-0.0001	role found
	47	$+0.679$ (0.989)	$+0.6516$	$+0.741$ $[+0.727, +0.753]$	1.000	$+0.0012$	role found

Result. At the decision block both checks pass, the pixel criterion included ($+0.0094$), and the label shuffle is clean. The role estimate is $+0.045$ bits with its whole interval inside ± 0.10, so by the rule specified in advance the pairwise readout carries negligible role information. Two facts go with that verdict. The same probe classifies role at 0.878 accuracy (0.961 at block 35), so the low bit count does not mean the roles are inseparable; the diagnostic below traces the gap to the probe's confidence. And the pairwise probe is weak in general: on the easy validity check its accuracy only just exceeds the majority rate, and at blocks 41, 44 and 47 it fails the pixel criterion. On identical rows with the same setting, the pooled readout carries the signal at every block ($+0.70$ to $+0.76$ bits, accuracy 1.000).

Why bits and accuracy disagree. Run after the analysis closed, this changes no verdict. Refitting the identical pipeline and recording the probabilities reproduces the frozen estimate ($+0.0448$ bits) and shows an under-confident probe: on held-out rows it gives the true class an average probability of 0.516 while choosing it correctly 87.8% of the time. A probe whose confidence matched its hit rate would gain $+0.465$ bits against a chance baseline, so this one realizes about a tenth of what its accuracy implies, where the pooled readout on the same rows realizes about three quarters ($+0.738$ of $+1.000$). One temperature fitted on training rows only raises the held-out estimate to $+0.60$ bits at block 38 and $+0.87$ at block 35. The verdict stands, but it describes this readout at $C = 10^{-4}$, where the bit criterion is conservative, and is not evidence that the endpoint tokens carry no role information. The calibration audit of Appendix A.19 flags exactly these two estimates and no estimate outside the endpoint readouts. The earlier version of this readout, at the same C, is the other place the audit predicted the failure (0.747 accuracy at $+0.029$ bits in Qwen-Image, 0.698 at $+0.019$ in Stable Diffusion 3.5); its activations could not be refit.

What this settles. Keeping the arrow's two ends apart, in this form, does not recover in bits the role signal the pooled readout carries. It does not show that the endpoint tokens lack role information, since the accuracy suggests some is present, nor that a differently tuned pairwise probe would fail. Answering either would take the kind of choice made after seeing the data that this analysis was built to exclude, so we leave both open.

A.15 TRANSPLANTING THE REMOTE ARROW (EXPLORATORY)

Cutting attention shows that the readout region needs *access* to the remote arrow, not what it takes from it. To ask that, we built a donor image identical to the original except that both lanes' remote arrows are reversed, which exchanges the two roles while leaving both target arrows pixel-identical, and wrote the donor's activations over the recipient's at the remote arrows' tokens during the forward pass. A probe fitted on unpatched runs, with the recipient's labels, was then applied unchanged. The patched region is each remote arrow's whole box rather than its head, because reversing an arrow moves its head and the two heads occupy different tokens; and one probe is used rather than two, because the donor's labels are the exact complement of the recipient's, so comparing two probes would compare x with $1 - x$ by construction. These runs use the 56 four-node diagrams and three of the six layouts.

With depth, the prediction stops following the recipient and follows the donor: accuracy against the recipient's own labels is 1.000, 0.820, 0.362, 0.003 and 0.000 at blocks 35, 38, 41, 44 and 47. Writing a run's own activations back changes nothing, as it should.

Why this is exploratory. The label shuffle missed zero at every block, in the unpatched condition too, so the estimator was offset with no intervention at all and the bits are unreadable at that setting. The cause, found afterwards, is the fourth entry of Appendix A.18: this analysis ran at a probe rank of 64 rather than the fixed 8. An accuracy that moves from 1.000 to 0.000 does not depend on a bits estimate, and a test that permutes the labels within each diagram puts the observed accuracy beyond all 2,000 permutations at every block. We report that as exploratory support, not a confirmed result. Two further limits: both lanes' arrows are reversed together, so this shows that the content of that region decides the prediction without isolating one lane; and all five blocks are patched in one pass, so the change across depth is cumulative and does not locate where the dependence forms.

A.16 REAL TEXTBOOK DIAGRAMS (AI2D-RST)

Every diagram above comes from our own code, and a fair worry is that the results depend on how clean and regular those images are. We therefore ran the same kind of measurement on real diagrams from school science textbooks, using AI2D-RST (Hiippala et al., 2021), which adds expert graph annotations to the AI2D collection (Kembhavi et al., 2016).

Choosing diagrams and arrows. We kept the diagrams in which every annotated connection is a directed arrow, 360 of the 1,000, of which 349 gave usable images. AI2D outlines every arrow, so in each diagram we chose the two arrows most alike in size and direction. The pairs are close, with a median size difference of 3% and a median angle difference of about 2°, but they are otherwise ordinary arrows: they connect different objects, carry different labels and sit in different places. They are matched in appearance, not in role.

Removing an arrow. A textbook diagram exists only as an image, so it cannot be redrawn without an arrow. We fill the arrow's outline with the surrounding background color and soften the edge slightly. Each diagram gives three images, the original and one with each arrow removed (Figure 12). The fill is an estimate: it can erase part of nearby text (first row of the figure) or leave a faint outline around a brightly colored arrow (third row).

What we measured. Qwen-Image at block 32 and $s = 0.2$, with two noise seeds and the same short prompt for every image ("a labeled science diagram"), so that the text cannot hint at the edit. The probe reads the tokens around each arrow; a real diagram has one layout, and folds hold out whole diagrams. We asked two questions: can the activations show *where* an arrow was removed, and can they tell *which* of the two was removed beyond what the arrow's position reveals? The second is not the role question. Two arrows in a real diagram connect different things and look different up close, so a positive result could rest on any of those differences; a negative one still shows that finding nothing beyond position is not a quirk of our own images.

What we found. The centroid of the activation change, with no probe fitted, locates the removed arrow to within about 2% of the image size (Table 15). Which arrow was removed is not recoverable: position alone predicts it better than chance, since the two arrows sit in different places, and adding

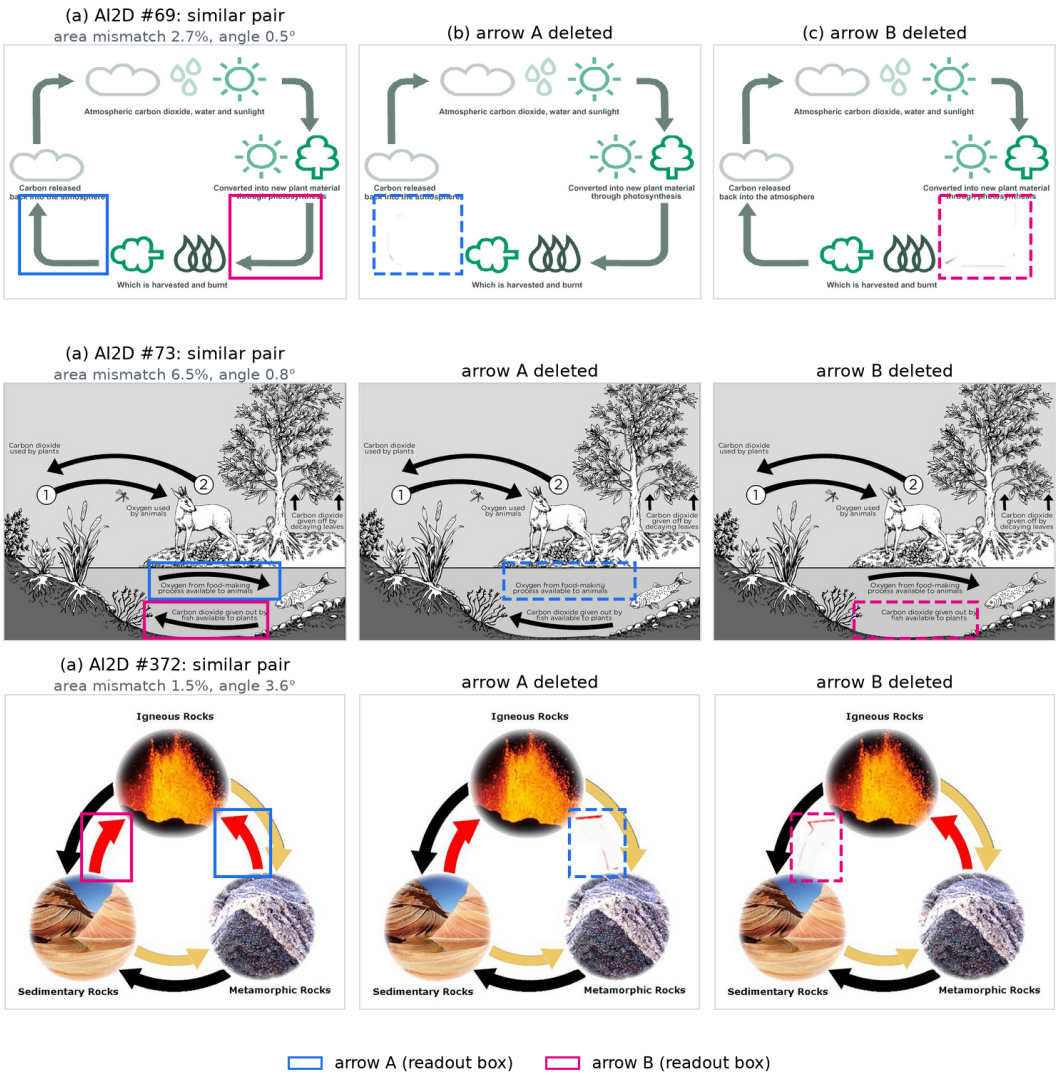

Figure 12: Three textbook diagrams from AI2D-RST. (a) The original, with the chosen pair of arrows boxed; each box is the region the probe reads. (b)–(c) The same diagram with arrow A or arrow B filled in with the background; the dashed box shows where it was.

Table 15: Results on 349 AI2D-RST diagrams. Qwen-Image block 32, $s = 0.2$, deletion edits, two noise seeds.

Measure	Value
Where the arrow was removed: median error of the activation change's centroid,	
as a fraction of image size	0.017
guessing the image center	0.322
Which arrow was removed: ΔS (bits)	-0.030
90% confidence interval	$[-0.040, -0.019]$
accuracy with activations	0.492
accuracy from position alone	0.585
label-shuffle ΔS (bits)	-0.0017

the activations makes held-out predictions slightly worse. The whole interval lies inside the ± 0.10-bit margin, so the result is negligible, and the shuffled-label check sits slightly below zero, small beside the margin. This uses one model, one block, one noise level and deletion edits only, on

arrows matched in size and direction and removed by painting them out; we read it as support for the main result on real diagrams, not as a second test of role.

Part VI The Record and Every Number

What was specified in advance, five measurements that passed for the wrong reason, the audit of the estimates inside the margin, the twin diagrams and two stricter splits, and the full tables. Appendices A.17–A.21.

A.17 Plan Record

Analyses we call *specified in advance* were entered in an append-only plan file before the run they govern. Each entry fixes the stimuli, the readout, the probe, the checks and the reading of every outcome, records the SHA-256 of the scripts it freezes, and is not edited afterwards; results are added beneath it as entries of their own. Two conventions matter for Table 16. An entry written after its numbers were seen says so in its own heading rather than being back-dated. And an analysis whose control fails is reported as failing and not re-tuned, which is why one row is exploratory.

Table 16: Reported analyses and their status in the plan file.

Analysis	Status	Note
Main grid and the denser block sweep	Specified in advance	Probe settings, checks and margin fixed in the first entry; the denser blocks added before they ran
Four-node against five-node split	Specified before it ran	Entered after the signal was seen
Standard probing baselines	Specified in advance	Rules and one prediction fixed before any baseline was computed
Redrawing the remote arrow	Specified in advance	Each set of versions entered after the previous set was probed and before it ran
Cutting attention (Qwen-Image)	Design specified in advance	Entry written after the run, and headed so
Cutting attention (Stable Diffusion 3.5, stretched shaft)	Specified in advance	Written before the run
Graded cut	Specified in advance	Levels and readings fixed before extraction
Pairwise endpoint readout	Specified in advance	Stopping rule fixed before extraction; the calibration diagnostic ran afterwards
Direction at the remote arrow	Specified in advance	Readout region corrected before extraction, and the correction recorded
Transplant	Specified in advance; control failed	Reported as exploratory; not re-tuned
Two-arrow family	Specified in advance	One extraction, one analysis; distance control and one-arrow ceiling fixed with it
Direction at both arrows	Specified in advance	Shuffle flags reported; no cell refitted
Loop-versus-skip family	Specified in advance	Reading fixed for each of three outcomes
Shifted and bent	Specified in advance	Entered together, before any probe ran
Branching	Specified in advance	The result was neither outcome named in advance
Direction at each branch	Specified before it ran	Entered after the branching result was known
Calibration audit	Specified in advance	Rule frozen before its one run
Twins in one fold; system held out	Not specified in advance	Run after the twin structure was found; probe settings unchanged, original cell reproduced first
Lane-separated cuts	Not specified in advance	Run after the joint cut; the both-heads version reproduces it bit for bit
Textbook diagrams	Outlined in the project plan	Selection and readout not fixed in advance; support only

A.18 Measurements That Passed for the Wrong Reason

This paper argues that a measurement can be correct on its own terms and still not mean what it appears to. That applies to our own instruments, and we record five cases because they are the same failure in miniature. None changes a conclusion we report: the first three never reached a reported result, the fourth is why the transplant of Appendix A.15 is exploratory, and the versions affected by the fifth are replaced by a rerun.

A readout region that encoded the answer. To ask whether an arrow's direction is present where it is drawn, the natural region to read is the arrowhead. It is useless: the arrowhead's *position* is the label. Across all 672 lane observations of the four-node corpus, the head's center sits at token column 44 when the remote arrow points outward and 41 when it points inward, with no overlap and no variance. The position-only probe would score perfectly, the gain would be zero by construction, and every condition would come back unreadable whatever the model encodes. The arrow's whole edge region sits at the same position for both directions, and a position-only probe on it runs at chance. Measuring the region's geometry before extracting anything caught this.

An occluder that moved with the label. The graded-exposure versions cover part of an arrowhead with a patch drawn from the tip backwards. The covered region's center therefore sits 27 to 37 pixels to one side of the center of the arrow's region when the arrow points outward, and exactly as far to the other side when it points inward; the version that places the same patch past the tip moves it 66 pixels. A manipulation meant only to remove evidence also moved it, in a direction that tracks the label. We quote no direction-at-the-arrow result from those versions; the one we report (Appendix A.7) comes from a version with no patch.

A control that passed because it was weak. The label shuffle asks whether the estimator's zero sits where it should, and the obvious test is whether the shuffled interval straddles zero. That test rewards imprecision: with few independent diagrams the interval is wide, so it straddles zero even when the offset is large, and a cell is certified *because* it is underpowered. A trial run did exactly this, passing an offset of -0.19 bits. The rule we use instead decides on the offset's size and on its size relative to the signal (Section 3.5), and records the straddle only for comparison with earlier analyses.

An estimator that changed while appearing not to. The probe's rank is set in the analysis code as the default value of an argument. A later script re-implemented the fitting routine with its own default rather than calling the original, which changed the rank from 8 to 64 while every line of the calling code looked unchanged. At the higher rank the probe reported a strong result at every block with a displaced zero; at rank 8 the same activations reproduce the reference to within 0.0004 bits with a clean shuffle. The trap has a second level: reassigning the module's constant changes nothing, because the default is bound when the function is defined, so a first attempt to vary the rank varied nothing and returned identical numbers four times over. Analyses here call the original routine and assert its settings at run time.

A version that was not what its name said. Two runs of the tighter-gap version, with the same name, the same stimulus definition, and the same model and settings, disagreed by 0.92 bits. Role alone cannot tell a contradictory measurement from a measurement of the wrong picture; validity can, because it is a property of the image. The first run matched the standard, uncrowded diagrams on every quantity (validity 0.7833 against 0.7835, role $+0.9351$ against $+0.9351$), so its two tighter-gap versions had been drawn at the standard spacing. The second run's validity is lower (0.558), as crowding predicts, and it is the run we report. The first run's other two versions drew their own images: the shifted icon's validity differs from the standard (0.746), and the stretched shaft is reproduced by an independent run ($+0.019$ both times). Both shuffles were clean, so no check caught this; comparing validity did.

A.19 Calibration Audit of the Estimates Inside the Margin

Supports: Section 4.7; the Reproducibility Statement.

A null in bits can be produced by a probe that separates the two classes and states its confidence too cautiously. Such a probe scores few bits while discriminating well, and the equivalence margin would then certify a real effect as negligible. This appendix reports a check for that failure across every estimate whose whole interval falls inside the margin. It produces a flag; no reported ΔS is re-scored, and every verdict stands as first recorded.

The rule, specified in advance. A cell is flagged if and only if both conditions hold. *Condition A, it discriminates:* $\mathrm{gap}_A = \mathrm{acc}_{\text{role}} - \mathrm{acc}_{\text{position-only}}$ has a 90% bootstrap interval over base diagrams

that excludes zero, *and* $\mathrm{gap}_A \geq 0.05$. Both halves are required, because an interval's relation to zero rewards precision rather than size. Discrimination is judged against each cell's own position-only probe, never against chance, since ΔS itself is measured against that probe. *Condition B, it is under-confident:* wasted $= L_{\text{role}} - H(\mathrm{acc}_{\text{role}}) \geq \delta$, where L_{role} is the probe's mean held-out log loss in bits and H is binary entropy, so $H(\mathrm{acc})$ is the log loss the same probe would have if its stated confidence matched its own hit rate. Condition B compares the probe only against itself; a calibrated probe sits at zero and an over-confident one is negative.

The threshold in condition B is δ, the equivalence margin itself, and deliberately so. A probe whose miscalibration wastes δ bits is wasting as much as the smallest effect the margin treats as meaningful, which is exactly the condition under which an effect of reportable size could sit in that cell and be scored as null by the waste alone. Both L_{role} and $H(\mathrm{acc})$ are log losses in bits, on the same scale as the two terms of ΔS, so the comparison is commensurable; a conversion of accuracy into bits against chance would not be. Condition A alone would flag small real effects the bit criterion scored correctly as small; condition B alone would flag probes that are under-confident *and* genuinely unable to separate the classes, where the null is real and under-confidence conceals nothing.

The rule, the scope and the cells in which we predicted flags were written into the plan file and frozen before the audit ran, and the audit was run once. We will release the frozen document and the cell inventory with their SHA-256 digests, which the audit script recomputes at run time and embeds in its output.

What it found. Of 361 estimates inside the margin, 147 could be refit. Two were flagged, and both are the endpoint readouts of Appendix A.14, the four endpoint-readout estimates we had predicted would flag. No estimate outside that set was flagged. (We say *estimate* here, not *cell*: a cell in Appendix A.2 is one model, edit and noise level, and holds many estimates.)

Estimate	ΔS	acc	$\mathrm{acc}_{\text{pos}}$	L_{role}	wasted
`pair_b35`	$+0.058$	0.961	0.500	0.942	$+0.705$
`pair_b38`	$+0.045$	0.878	0.500	0.955	$+0.420$

Both probes assign a mean probability of about 0.52 to the class they choose while being right in 88–96% of held-out cases. The refits reproduce the stored ΔS exactly.

The rule is not a formality that passes everything. Condition A holds in 60 of the 147 estimates, so most of the discriminating ones are stopped by condition B rather than by the accuracy floor; 58 clear A and fail B, and 56 of the 147 have *negative* waste, meaning the probe is over-confident rather than under-confident and could never flag. All 95 estimates from the whole-arrowhead attention-masking runs and the occlusion runs of Appendix A.7 were refit and none was flagged, which is the check that bears on the causal claim of Section 4.4. The graded-occlusion and distance-sweep redraws are not among them; they are in the unreachable set below. The remaining shifted, bent and branching estimates (12) and 22 of the 28 two-arrow estimates were also refit, and none was flagged; the other six two-arrow estimates are the lane-kind readouts excluded by design, and the 45 partial-cut estimates are in the unreachable set.

What the audit could not reach, and why. The remaining 214 estimates are reported as unaudited; they are not counted as having passed. They divide as $135 + 11 + 50 + 18$. Activations were not retained for a number of earlier runs. Where the stimuli and parameters allowed it we re-extracted them into separate directories, leaving the originals in place, and against the surviving item lists those re-extractions give identical row sets and reproduce every stored estimate to five decimal places on 64 of 64 cells. Twelve further extraction directories were not recoverable in time, which accounts for 135 estimates, among them the graded-occlusion and distance-sweep redraws. Another 11 failed their refit for reasons other than missing activations: six three-way lane-kind readouts, which are not role estimates and were excluded by design, and five loop-versus-skip estimates whose arm name the refit path does not recognize. 50 estimates in the earliest block scans have no refit path; and for the 18 from the first phase of this project the activations are gone with no stimulus record to rebuild them from, so condition B is not computable there at all.

Two consequences should be stated rather than left to inference. First, "no flag outside the predicted set" is a statement about 147 estimates, not 361. Second, two of the four estimates in the predicted

set are themselves among the unreachable ones: `expP_qwen_pooled` and `expP_sd35_pooled` have stored accuracy gaps of 0.247 and 0.198, resembling the two that flagged, so the prediction was confirmed on half of the set it named.

Disclosure. Sixteen estimates in the two-arrow family were visible before the freeze, while a refit path was being validated; all sixteen were not flagged. No threshold and no formula changed afterwards: the values 0.05, 90% and δ, and the expression $L_{\text{role}} - H(\text{acc})$, are those the audit script carried when it was written, before any cell was refit. The frozen document initially described condition B by a different quantity, the gap between accuracy and mean confidence; the freeze corrected the document to match the code rather than the reverse, and records that it had been wrong. Three refit paths were added before the run, each gated on reproducing the stored ΔS: one for the two-arrow family, one routing a cell to its re-extracted directory, and one restricting a cell to the three layouts it was run on rather than six, which changes the position-only baseline and is a correction rather than a convenience.

A.20 Twin Diagrams and Two Stricter Splits

Both role lanes of a diagram carry the same icons and labels, so the variant factor that puts the farther-continuing role on lane A or lane B repeats the lane orderings: a variant with that role on lane A, drawn in the order (A, B, C), is the same picture as its partner with the role on lane B, drawn in (B, A, C). Every base diagram therefore has a *twin* whose six images are its own in another order. The stored activations confirm it: all 4,032 rows of the dense Qwen-Image cell have a bit-identical twin row with the same label and position features. The 112 base diagrams are 56 distinct image sets, and the folds of Appendix A.4 left the twin in training for 44 of the 56 four-node diagrams (82 of 112 overall).

We refit the same estimator two ways, after the plan and with nothing tuned (rank 8, $C = 0.003$). *Twins in one fold* puts each pair in a single fold, and its intervals resample the 28 four-node twin pairs. *System held out* uses seven folds, one per scientific system, and excludes every variant, layout and seed of that system from the scaler, the principal components and both probes; twins share a system, so this split cannot leak them. Block 38 was chosen with every system in view, so we also choose the block inside each fold, by the same split over the six training systems. Before either refit, the same code with the original folds reproduced the stored four-node cell exactly ($+0.9352$ bits, accuracy 0.9918).

The four-node signal at blocks 35 and 38 survives both splits (Tables 17 and 18). Blocks 41 to 47 depend on the folds: with twins in one fold, block 44 falls from $+0.404$ to $+0.222$ and block 47 from $+0.789$ to $+0.615$, and with a system held out their accuracy varies from system to system (0.604 to 0.896 at block 44). Intervals over base diagrams count each twin pair twice and are about $\sqrt{2}$ too narrow: at block 38 the interval over twin pairs is $[+0.911, +0.957]$, against the $[+0.918, +0.951]$ of Section 4.1. The two-arrow family, refit the same way, is reported in Appendix A.9; the other families, which share the construction, were not refit.

Table 17: Four-node role signal under three splits. Qwen-Image, reversal, $s = 0.2$, six layouts. ΔS in bits, accuracy in brackets, position-only 0.500 throughout. Twin intervals are 90% over the 28 twin pairs. For the system-held-out split, the mean accuracy over the seven held-out systems and the lowest. Every shuffle in the first two columns is clean; Table 18 gives the system-held-out ones at block 38. Run after the plan.

Block	Original folds	Twins in one fold	System held out: mean (lowest)
35	$+0.930$ (0.987)	$+0.920$ $[+0.897, +0.941]$ (0.993)	0.994 (0.969)
38	$+0.935$ (0.992)	$+0.925$ $[+0.898, +0.950]$ (0.990)	0.996 (0.969)
41	$+0.498$ (0.829)	$+0.447$ $[+0.363, +0.535]$ (0.839)	0.832 (0.646)
44	$+0.404$ (0.812)	$+0.222$ $[+0.193, +0.253]$ (0.731)	0.799 (0.604)
47	$+0.789$ (0.949)	$+0.615$ $[+0.556, +0.675]$ (0.897)	0.894 (0.812)
38, all 112	$+0.381$ (0.794)	$+0.394$ $[+0.329, +0.457]$ (0.806)	0.836 (0.667)

Table 18: Each scientific system held out in turn, four-node diagrams, block 38. 192 scored rows per system (96 distinct, because of twins); position-only 0.500. Accuracy interval is 90% over the system's base diagrams. The last column chooses the block inside the fold, without the held-out system. * label-shuffle flag (offset -0.016, a sixtieth of the estimate).

Held-out system	Accuracy	ΔS	Shuffle	Block chosen in fold (accuracy)
Circuit	1.000 $[1.000, 1.000]$	$+0.984^*$	-0.016	38 (1.000)
Feedback	1.000 $[1.000, 1.000]$	$+0.975$	$+0.005$	35 (1.000)
Food web	1.000 $[1.000, 1.000]$	$+0.962$	$+0.001$	38 (1.000)
Growth chain	1.000 $[1.000, 1.000]$	$+0.985$	$+0.008$	35 (1.000)
Heating	1.000 $[1.000, 1.000]$	$+0.938$	$+0.005$	47 (0.979)
Light response	1.000 $[1.000, 1.000]$	$+0.976$	$+0.006$	38 (1.000)
Matter cycle	0.969 $[0.948, 0.990]$	$+0.842$	-0.002	38 (0.969)
All seven	0.996			0.993

A.21 FULL RESULTS

The main text reports the few numbers each argument needs. This section gives the full tables behind them, all read directly from the saved results. Every value is role ΔS in bits from the linear probe, and marks follow the label-shuffle rule of Section 3.5, rescored from each result's stored offset: a star for a flagged estimate, a dagger for one whose offset is too large to read.

The full grid. Table 19 lists the value in every cell of the planned grid: every model, edit type, noise level and depth tested on the 112-diagram corpus. Only one row rises above the ± 0.10-bit margin: Qwen-Image with reversed arrows at $s = 0.2$, from block 35 onward.

Table 19: Role signal (ΔS, bits) for every tested cell, 112 diagrams, linear probe. * label-shuffle flag; † offset too large to read (Section 3.5); — validity check failed (unreadable); blank: not extracted.

Relative depth	15%	30%	55%	60%	65%	70%	75%	80%
Qwen-Image / SD3.5 block	9/6	18/11	32/20	35/22	38/24	41/26	44/28	47/30
Qwen-Image / reverse / s=0.2	+0.002	+0.002	+0.000	+0.266	+0.381	+0.238	+0.167	+0.253
Qwen-Image / reverse / s=0.4	+0.000	+0.000	+0.000	+0.002	+0.003	+0.004	+0.008	+0.002
Qwen-Image / delete / s=0.2	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000	+0.000
Qwen-Image / delete / s=0.4	+0.000	+0.000	+0.000					−0.000
SD3.5 / reverse / s=0.2	−0.001	−0.000	+0.004	+0.019	+0.017	+0.014	+0.002	+0.002
SD3.5 / reverse / s=0.4	−0.001	−0.000	+0.001	+0.001	+0.002	+0.003	+0.001	+0.005
SD3.5 / delete / s=0.2	—	—	−0.000					−0.000
SD3.5 / delete / s=0.4	—	—	+0.000					+0.000

Four-node versus five-node diagrams. Splitting that row by lane length (Table 20) shows that the signal is carried by the four-node diagrams. In the late blocks they reach $+0.40$ to $+0.94$ bits, while the five-node diagrams never exceed $+0.02$. This is why the source experiments in Appendices A.7 and A.8 use the four-node diagrams. The five-node null is not a test of hop count: those diagrams draw the remote arrow between closely spaced icons, which removes the signal in four-node diagrams as well (Appendix A.7). The diagrams of Appendix A.9 are also five-node, but draw every arrow at the four-node corpus's widest spacing, and there the pairs separated by the last arrow reach $+0.96$ bits at block 44. That family also changes the role rule, so it supports the crowding reading rather than settling it.

Table 20: The Qwen-Image signal by lane length (reversal, $s = 0.2$, six layouts). Nearly all of it comes from the four-node diagrams. * label-shuffle flag; † offset too large to read (Section 3.5).

	All 112		Four-node (56)		Five-node (56)	
Block	ΔS	Acc.	ΔS	Acc.	ΔS	Acc.
9	+0.002	0.541	+0.006	0.570	+0.000	0.506
18	+0.002	0.529	+0.006	0.545	+0.001	0.516
32	+0.000	0.514	+0.000	0.501	+0.000	0.505
35	+0.266	0.756	+0.930	0.987	+0.004	0.529
38	+0.381	0.794	+0.935	0.992	+0.017	0.562
41	+0.238	0.732	+0.498	0.829	+0.015	0.564
44	+0.167	0.663	+0.404	0.812	+0.002	0.514
47	+0.253	0.724	+0.789	0.949	+0.017	0.557

Blocking attention, block by block. Table 21 extends the block-38 comparison in the main text to all late blocks. Blocking the remote arrowheads brings the estimate to within about 0.005 bits of zero at every block, while the size-matched cut over the control arrowhead leaves it close to the unmasked value. The shaft-stretched arm, which starts with no signal, stays inside the margin under both conditions; one of its cells, arrowheads blocked at block 47, has a shuffle offset too large to read in bits. In Stable Diffusion 3.5 every value is already inside the margin; blocking the arrowheads removes the small positive values of the deepest blocks, and the size-matched cut mostly does not.

Table 21: Blocking attention at every late block (four-node diagrams, three layouts). Top: Qwen-Image. "Re-probe" fits the unmasked activations on the same three layouts without the masking code; it matches the empty mask, so the code itself changes nothing. Validity accuracy is the range over the three conditions. Each entry gives ΔS with the probe's role accuracy in parentheses; the position-only baseline is 0.464 on these three layouts (Table 9), so an estimate inside the margin whose accuracy stays near chance is null on both measures. Bottom: Stable Diffusion 3.5. * label-shuffle flag; † offset too large to read (Section 3.5).

	Standard arm					Shaft-stretched arm	
Qwen block	Re-probe	Nothing	Arrowheads	Size-matched	Validity acc.	Nothing	Arrowheads
35	+0.615 (0.905)	+0.614 (0.909)	+0.003 (0.540)	+0.621 (0.874)	0.931–0.969	+0.046 (0.631)	−0.002 (0.510)
38	+0.730 (0.938)	+0.730 (0.938)	+0.005 (0.525)	+0.636 (0.866)	0.951–0.980	+0.005 (0.516)	−0.004 (0.499)
41	+0.174 (0.710)	+0.174 (0.711)	+0.003 (0.527)	+0.146 (0.680)	0.948–0.959	+0.001 (0.501)	−0.006 (0.484)
44	+0.485 (0.854)	+0.485 (0.856)	−0.002 (0.503)	+0.436 (0.820)	0.916–0.945	−0.004 (0.499)	−0.006 (0.476)
47	+0.943 (0.999)	+0.943 (0.999)	−0.005 (0.476)	+0.946 (1.000)	0.903–0.953	+0.006 (0.551)	−0.000† (0.518)

SD3.5 block	Nothing	Arrowheads	Size-matched
22	+0.001 (0.516)	−0.006 (0.494)	+0.002 (0.536)
24	+0.009 (0.552)	−0.001 (0.509)	+0.013 (0.577)
26	+0.014 (0.558)	−0.000 (0.488)	+0.014 (0.567)
28	+0.011 (0.564)	−0.002 (0.491)	+0.013 (0.564)
30	+0.021 (0.571)	−0.001 (0.501)	+0.017 (0.552)

Partial blocking, block by block. Table 22 gives every point of the dose curves. In the standard arm the estimate falls steeply at every block. Blocking a quarter of the arrowhead removes about four fifths or more of the signal at blocks 35–44 and about half at block 47, and by half blocked almost nothing is left anywhere. At blocks 44 and 47 the last step rises by less than 0.003 bits between values that are already at chance. The shaft-stretched arm stays inside the margin at every block and level, with the same unreadable cell at full blocking in block 47; apart from block 35, where it starts at $+0.046$ bits, it varies by less than 0.01 bits.

Table 22: Partial blocking at every late block: role ΔS (bits) against the fraction of arrowhead tokens blocked (Qwen-Image, four-node diagrams, three layouts). The columns for 0 and 1 are the "nothing" and "arrowheads" conditions of Table 21. * label-shuffle flag; † offset too large to read (Section 3.5).

Block	$f = 0$	0.25	0.5	0.75	1
Standard arm					
35	+0.614	+0.126	+0.031	+0.010	+0.003
38	+0.730	+0.156	+0.030	+0.010	+0.005
41	+0.174	+0.037	+0.013	+0.005	+0.003
44	+0.485	+0.040	+0.002	−0.004	−0.002
47	+0.943	+0.449	+0.006	−0.006	−0.005
Shaft-stretched arm					
35	+0.046	+0.008	+0.004	−0.000	−0.002
38	+0.005	−0.004	−0.002	−0.004	−0.004
41	+0.001	−0.005	−0.005	−0.004	−0.006
44	−0.004	−0.006	−0.006	−0.006	−0.006
47	+0.006	+0.002	+0.005	−0.000	−0.000†