WHAT DOES A RECOVERED RULE EXPLAIN? DIAGNOSING WHICH RELATIONAL CONCEPTS A NEURAL MODEL COMPOSES

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a structural explainability method that builds partial, testable explanations of a neural model's decisions and audits where they hold. A symbolic program recovered from a model's decisions alone serves as an initial explanatory hypothesis; the audit then tests its relational constituents against hand-built scenes placed at each constituent's own decision boundary, verified with the task's executable ground truth. Where a constituent fails this test, an explicit alternative program offers a candidate account of the discrepancy, and we evaluate its predictive validity in a held-out case study rather than by how well it matches the errors it was chosen to explain. A linear probe further asks whether the information a constituent needs is at least present in the model's own representation. The result is an *explanation profile*: a record of which parts of a recovered explanation hold up, which are better accounted for by a named alternative, and which remain unexplained.

Applied to Kandinsky Patterns across eight relational tasks and three architectures, including a perception-free model given exact object descriptions, recovered programs agree with the generating rule on standard test examples but fail to predict the model at these targeted boundary cases. For one touching-and-counting task, an alternative that counts touching clusters rather than touching objects raises held-out agreement with the perception-free model from 45.7% to 61.0%; it does not transfer to either image-based model, and matching the model's decisions better than the recovered rule does is evidence about behaviour, not about what computation the model implements internally. Separately, per-object contact remains linearly decodable from the perception-free model's representation at 94.5% accuracy despite the model failing the composed task – a result the probe alone cannot resolve into use or disuse of that information. Rather than a single verdict on whether a recovered rule explains a model, the method produces a structured account of where it holds, where an alternative improves it, and where the evidence available from behaviour and representation runs out.

1 INTRODUCTION

Figure 1 shows a scene with four touching objects arranged in two separate clusters. The generating rule labels it `TRUE`, while a neural model receiving exact object descriptions predicts `FALSE`. A symbolic program recovered from this model's predictions agrees with the generating rule on standard test examples, yet disagrees with the model on this scene. Recovering a program that reproduces standard-distribution behaviour is a natural first explanatory hypothesis about the model; the disagreement shows that a hypothesis of this kind is not yet a finished explanation, and points to where it still needs to be tested.

This gap between agreeing with a rule and predicting a model is what the rest of the paper works out: where a recovered explanation holds, what replaces it when it does not, and what evidence beyond raw agreement can settle the question in either direction. A structural explainability method built around this idea decomposes a recovered program into relational constituents and audits each one where it is meant to matter – at its own decision boundary, against alternative accounts of what a model does when the recovered rule fails there, and against what its own representation makes available. Section 3 defines this audit; applied to Kandinsky Patterns, it contributes:

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.

- An audit that scores a recovered explanation's constituents at their own decision boundaries, compares named alternatives against held-out predictive performance rather than error overlap alone, and probes whether a constituent's information is present in the model's representation (Section 3).
- Boundary tests across eight relational tasks and three architectures, including a perception-free model given exact object descriptions, locating where a recovered explanation holds and where it needs revision (Section 4.1).
- A held-out-validated alternative that raises predictive agreement with a perception-free model's decisions but does not transfer to either image-based model, and a linear probe showing per-object contact remains decodable from that same model despite its failure on the composed task (Section 4, Appendix A.18).

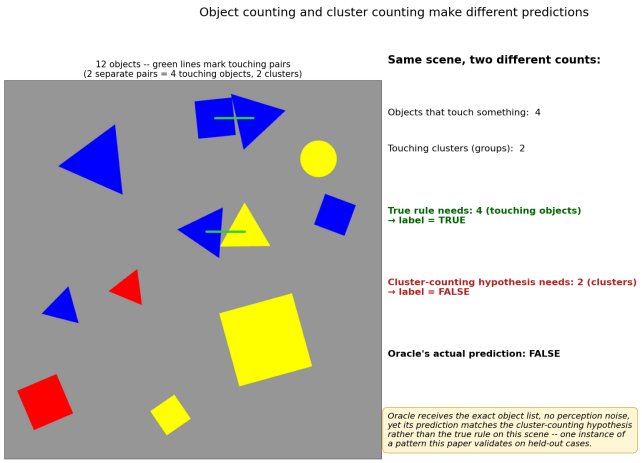


Figure 1: 4 objects touch something but form only 2 clusters. True rule: `TRUE` (objects). Oracle and cluster-counting hypothesis: `FALSE`.

2 TESTBED

Kandinsky Patterns [5] are synthetic scenes with an explicit, machine-checkable generating rule – geometric objects composed by a boolean or counting predicate, in the spirit of other synthetic rule-induction benchmarks such as KANDY [6]. We use 8 rules spanning a range of relational and counting structure (`left-red-implies-right-blue`, `radial-symmetry`, `touching-count-matches-red-left`, and five others; full list, definitions, and one illustrative scene per rule in Appendix A.20), plus two rules added later specifically to probe the audit's own limits (Section 4.1): a positive-control rule the audit should certify, and a rule motivated by a real clinical cutoff. Rules vary in whether they require deriving a new relational predicate from raw coordinates (e.g. contact, symmetry) rather than reading an existing attribute; the full five-axis complexity profile underlying this distinction, and the resulting rule-by-rule breakdown (Section 4.5), are in Appendix A.5.

Three model classes separate a perception confound from a composition failure: **VQVAE**, a VQ-VAE tokenizer feeding a relational Transformer; **SlotAttn-Tr**, a Slot Attention encoder feeding the same Transformer; **Oracle**, given the true object list directly, so boundary failure cannot be attributed to pixel-resolution or segmentation noise (confound-isolation ablation in Appendix A.9). A fourth model, **SlotAttn-RN**, swaps the shared Transformer for an attention-free Relation Network [2] and is trained as a self-attention robustness check on a subset of rules, not the full library (Section 4.5). All four are trained once, independently, per rule, at 3 seeds; no model is retrained to compute `CF`/`AM`/`E` or any profile label – all are scored by re-running frozen checkpoints. The one exception, scoped as its own separate check rather than part of this computation, is the geometry-corrected replication (Appendix A.15), which deliberately retrains every model from scratch on corrected data as its method.

3 THE STRUCTURAL EXPLAINABILITY METHOD

Our structural explainability method turns a program recovered from a model's decisions into a testable, per-constituent explanation: which relational sub-concepts the recovered program's decisions are consistent with, which are better accounted for by a named alternative, and which remain unexplained. Recovery itself uses only the model's decisions; auditing what it recovers draws on more – the task's executable ground truth, hand-built boundary scenes, and, where accessible, the model's own internal representation.

S_R and the interpreter h. A *rule-level structure* S_R is a semantically typed relational program (typed relations in Appendix A.4) executed by a simple, transparent interpreter h that maps (S_R, x) to a decision for scene x. S_R is recovered from a model's own decisions by program synthesis (Section 3), never from the true generating rule.

Two fidelity criteria. For scenes x, a model's own decision $f(x)$, and the true rule's decision $f^\star(x)$:

$$\texttt{ground_truth_fidelity}(S) = \Pr_x\big[h(S,x) = f^\star(x)\big],$$
$$\texttt{decision_fidelity}(S) = \Pr_x\big[h(S,x) = f(x)\big].$$

The first asks whether S agrees with the true rule; the second, whether it agrees with the model, right or wrong. A program can reach $\texttt{ground_truth_fidelity} = 1$ on the standard distribution while the model it was recovered from behaves nothing like that program at the rule's own boundary (Section 4.2); keeping these two questions separate throughout is the paper's central methodological commitment.

Constituent fidelity. A rule's minimal program decomposes into named constituents c – e.g. *contact*, *region*, *equality/threshold comparison*, *aggregation*, and *shape-dependence of contact* (is the pairwise predicate correctly conditioned on object identity, not just position). For model m and c's own targeted counterexample categories (95% Wilson interval):

$$\texttt{CF}(c, m) = \text{accuracy on the scenes targeting } c, \qquad \text{never on the standard distribution.}$$

$\texttt{ground_truth_fidelity} = 1.0$ alone – standard-distribution agreement with the true rule – does not indicate whether c passes its own boundary test; $\texttt{CF}(c, m)$ is deliberately never computed on that distribution for this reason.

Alternative computations and constituent encoding. When $\texttt{CF}(c, m)$ is low, we score a model's specific wrong decisions against named alternative computations a for c – a shape-blind distance threshold in place of true contact, say – restricting $\texttt{decision_fidelity}$ to the scenes where the model disagrees with the true rule: $\texttt{AM}(c, m, a)$ is the share of m's errors on c's scenes that a predicts, and a is flagged *not identifiable* wherever it coincides with the true rule or with another alternative. Separately, and only where a model's internal representation is actually accessible (our perception-free oracle), we train a linear probe on frozen activations for the ground-truth quantity that c needs, giving $\texttt{E}(c, m)$: whether that information is even linearly present, independently of whether m's own decision head uses it. A constituent can fail behaviourally, with CF low, while its information is well encoded, with E high; $\texttt{decision_fidelity}$ against the true rule alone cannot distinguish this case from the constituent never having been represented at all.

Explanation profile. Combining CF, AM, and E per constituent and model gives an *explanation profile*, assigned identically to every (rule, constituent, model) cell with no threshold adjusted after seeing a result. Table 1 gives the exact condition and permitted interpretation of each label. Where a candidate a is additionally calibrated and scored on disjoint held-out instances (only done for one rule in this paper, Section 4.3), we report the stronger *behaviourally consistent with* a instead of *high error overlap*, reserved for that tested case.

Scope. Every criterion above is *behavioural and representational*: it asks what a model's decisions and its linear representation are consistent with, not whether S_R is the model's own implemented computation in a mechanistic sense (Section 6). Two consequences recur throughout the audit: a

Table 1: Explanation-profile labels: the behavioural verdict is exactly one of the first four rows; the fifth is an independent flag that can co-occur with any of them, computed only where E is available (the perception-free oracle).

Label	Exact condition	Permitted interpretation
Passes the constituent boundary test	CF's 95% Wilson lower bound ≥ 0.90	The recovered program's behaviour on c's own boundary scenes is well supported; not a claim about internal implementation.
High error overlap with a	CF's 95% Wilson upper bound ≤ 0.50; exactly one alternative reaches $\texttt{AM}(c, m, a) \geq 0.80$, identifiable	a matches the model's errors on the scored sample – a descriptive fact about that sample, not a tested predictive claim.
Behaviourally inconsistent, no single alternative identified	Same CF condition; two or more alternatives reach AM ≥ 0.80	The errors are consistent with more than one candidate; none can be singled out from the fixed alternative set tested.
Unexplained	None of the above (including cells whose CF fails 0.90 without falling below the stricter 0.50 needed to score an alternative at all)	The available evidence does not meet the criteria for supporting the recovered program or identifying a single alternative on the tested boundary scenes.
Linearly decodable despite task failure	Same CF upper-bound condition; $\texttt{E}(c, m) \geq 0.85$ *and* above the probed target's own majority-class base rate (oracle only)	The needed information is linearly present in the representation; says nothing about whether the model's own decision head uses it.

constituent's CF score localizes *where* the recovered program stops predicting the model, not *why* the model itself errs; and a model's errors being behaviourally consistent with a named alternative a is evidence a predicts the model better than the recovered program on the cases tested, not evidence the model internally implements a's computation.

Audit. The following four-step procedure applies CF, AM, and E to a recovered S_R in practice.

Step 1 – behavioural program recovery. $f(x) \rightarrow S_R$: a typed relational DSL (grammar in Appendix A.3) is searched by enumerate-and-score program synthesis against a model's own decisions on held-out scenes, with no ground-truth access during search – only after a candidate is fixed is it scored against both `decision_fidelity` and `ground_truth_fidelity`.

Step 2 – per-constituent counterexamples. For each constituent c of a recovered S_R reaching $\texttt{ground_truth_fidelity} = 1.0$ on the standard distribution, we hand-build scenes placed directly at c's own decision boundary (e.g. the narrowest non-touching gap, an exact-count equality broken by one object), verified against the rule's own executable evaluator before scoring (exact construction for every category in Appendix A.8).

Step 3 – score against the true rule and named alternatives. Each model's decisions on c's scenes give $\texttt{CF}(c, m)$; where a model disagrees with the true rule, its specific wrong decisions are additionally scored against the named alternative computations for c, giving $\texttt{AM}(c, m, a)$. Alternatives are not invented anew per rule: every one of the 19 named alternatives used across our 10 rules (Appendix A.6) is an instance of one of five AST-level edit types applied to the recovered program: drop a conjunct, shift a threshold constant, widen a tolerance parameter, change a comparison operator, or change an aggregation function – a small, fixed operator set applied uniformly across rules.

Step 4 – constituent probes. On the oracle's frozen checkpoints only (no perception stage to introduce nuisance variation), we train a linear probe at each representation depth to decode the ground-truth quantity each constituent needs. Two distinct senses of "held out" apply here: the probe draws on the same scenes used in Step 2, which were never used to train the oracle model itself, and within that pool the probe is fit on a 70% split and scored on the disjoint 30% it never sees during fitting – giving $\texttt{E}(c, \text{oracle})$ independently of whatever CF and AM say about the oracle's own decisions.

4 RESULTS

The results below take up the three questions from the introduction in order: whether a recovered rule survives targeted boundary cases (Sections 4.1–4.2), which alternative computation matches a model's behaviour where it does not (Section 4.3), and whether the needed information is nonetheless represented (Section 4.4). Section 4.5 then checks the result against training exposure and architecture; an attribution-method comparison is in Related Work (§5). Figure 4 in the appendix gives the full picture – every (rule, constituent) cell, every model – before the per-rule detail below unpacks it.

4.1 FIT DOES NOT CERTIFY BOUNDARY PERFORMANCE

Across all 8 rules and all three models, standard-distribution `ground_truth_fidelity` sits at or near 1.0 (Appendix A.7) – program-vs-true-rule agreement, not model accuracy – yet this fit does not predict whether a constituent passes the boundary test built to stress it specifically: none of the cells on rules with high standard-distribution model accuracy do. On `parity-five-indicators` the two diverge: all three models' own `decision_fidelity` against the ground-truth-exact recovered program is only 0.513–0.527, near chance, so its boundary results below reflect a task never learned on the standard distribution either, not a model-faithful explanation collapsing. Once every boundary-adversarial category is pooled onto its mapped constituent (Appendix A.6), across all 8 rules' 48 (rule, constituent, model) cells – including `parity-five-indicators`' 3, kept in for a complete overview – no cell reaches the chosen certification threshold, a 95% Wilson lower bound of at least 0.90; the closest any cell comes is an *observed* accuracy of 0.967, whose own 95% Wilson lower bound is only 0.886 – insufficient evidence to certify at the sample size scored, not proof of collapse for that cell specifically. The claim that a previously model-faithful explanation collapses at the boundary is restricted to the 45 cells on the 7 rules with high standard-distribution model accuracy (excluding `parity-five-indicators`, which never had a model-faithful explanation to collapse); on that subset, a standard-split fit of 1.0 is compatible with none of these 45 cells reaching the certification bar for the constituent in question.

That bar is not simply unreachable. We added a ninth rule, `majority-red-of-four` (count(red) ≥ 3 of exactly 4 objects, requiring no constructed relation), specifically to check whether the audit can certify success and not only detect failure. On its own two boundary categories, the oracle reaches `CF` $= 1.000$ – the only cell in the paper that passes the constituent boundary test. VQVAE and SlotAttn-Tr do not (0.808 and 0.633; SlotAttn-Tr's own sanity-baseline accuracy here, 0.917, is itself the first validity-gate reading below our 0.95 bar, so we flag its boundary numbers with that caveat). VQVAE's errors are 82.6% attributable to `H_ge2`, "≥ 2" standing in for the true "≥ 3", past our identifiability bar – yet its own `CF` still does not cross that threshold.

A tenth rule adds an additional synthetic task motivated by Ki-67 scoring: a proliferation-index cutoff from digital pathology, 20%, following the 2013 St Gallen consensus [4] as one representative choice among several guideline-dependent thresholds that range roughly 10–25%. All three models clear the validity gate here, but they diverge sharply – and in opposite directions – right at the boundary itself (`CF` $= 0.483, 0.517, 0.658$, all *unexplained*): VQVAE and SlotAttn-Tr need *more* red triangles than the true threshold before they will say True, while Oracle needs *fewer* (Appendix A.6). This is a single rule, not a statistically powered claim across the space of clinical cutoffs: it rules out the collapse being an artifact of one synthetic construction, without showing it holds at every possible cutoff.

4.2 BOUNDARY COLLAPSE: `TOUCHING-COUNT-MATCHES-RED-LEFT`

All three architectures recover the true program exactly (`ground_truth_fidelity` $= 1.0$, zero variance, 3 seeds) – matching the generating rule on standard test examples while, as the table below shows, failing to predict the model at targeted boundary cases. Boundary-adversarial scenes targeting five constituents (contact, region, exact-count comparison, aggregation, and shape-dependence of contact; Appendix A.8) reveal substantial, constituent-dependent discrepancies:

Constituent (CF)	VQVAE	SlotAttn-Tr	Oracle	SlotAttn-RN
contact, tight-true vs. near-miss (n=120)	0.500	0.500	0.500	0.500
region (n=240)	0.396	0.242	0.750	0.258
equality/threshold comparison, false side (n=60)	0.000	0.000	0.083	0.017
equality/threshold comparison, hard-true side (n=60)	0.983	1.000	0.900	1.000
aggregation (n=120)	0.442	0.025	0.000	0.042
shape-dependence of contact (n=537)	0.750	0.771	0.196	0.695

contact's 0.500 is a symmetric two-sided boundary result (`touch_boundary_true` is itself the tightest-possible still-touching case, not an easy true case). **equality/threshold comparison** pools a sanity baseline (1.000 for all models) with one boundary category, reported separately since pooling mechanically drives CF toward 0.5; the **hard-true side** row (exactly as adversarial as the false side, restoring the count-equality instead of holding it fixed) tests the same way. The two rows together show a one-directional miscalibration: every model needs *less* evidence than the rule requires to say True, near ceiling on the equally-hard true construction while collapsing on the false one, with none of the five named alternatives explaining the false-side errors (≤ 0.036 pooled attribution). Full numbers and a plot of the same collapse are in Appendix A.8 (Figure 6).

4.3 Alternative Programs Provide Partial, Model-Specific Accounts

On `touching-count-matches-red-left`, we select an alternative using a calibration subset and evaluate it – with no further fitting – on disjoint holdout instances from the same construction families, to ask whether it predicts the model better than the recovered rule does. For the perception-free model, counting touching *clusters* rather than touching objects increases agreement across *all* held-out decisions from 45.7% to 61.0%. For VQVAE and SlotAttn-Tr, the same alternative *reduces* agreement, from 63.2% and 59.5% to 29.3% and 27.9%, respectively (Table 2): cluster counting is a partial account of the perception-free model's behaviour that does not transfer to the image-based models.

Table 2: Holdout `decision_fidelity`: agreement with *all* model decisions, not only errors, on instances never used to select the alternative.

Hypothesis	VQVAE / SlotAttn-Tr	Oracle (perception-free)
H_{true} (recovered rule)	0.632 / 0.595	0.457
H_{cluster}	0.293 / 0.279 (worst of 6)	0.610 (best of 6)

Error-only matching requires additional care. In binary classification, an alternative matches a wrong model prediction exactly when it also disagrees with ground truth; consequently, a high match rate can arise from the construction of the test set alone, not from anything the alternative gets right about the model. We therefore report full decision agreement (above) alongside error matching. For the perception-free model, the alternative matches 82.0% of held-out errors, compared with its own 73.8% disagreement rate with ground truth on the pooled evaluation scenes – a real but modest lift over that baseline, not the far larger gap a flat 50% chance floor would suggest. This association supports further investigation, but neither identifies the implemented mechanism nor establishes generalization beyond the tested construction families. The corrected significance test against this baseline, the full catalogue of alternative-hypothesis results across all eight rules, and the profile heatmap (Figure 3) are in Appendix A.18 and Appendix A.6.

4.4 Decodable Primitives Despite Task Failure

Linear probes on the oracle's frozen representation decode per-object `touches(o)` at 94.5% and `in_left_region(o)` at 99.3%, both above the 0.85 threshold set for `E` – so this information is linearly present in the representation even where the model fails the composed task. The per-scene counts these decisions are actually supposed to rest on tell a different story, though: touching-*object* count is decodable only at 23.1%, while touching-*cluster* count reaches 83.9%. The oracle's shape-dependence-of-contact constituent is therefore both *behaviourally consistent with H_cluster* and *linearly decodable despite task failure* at once: per-object contact remains decodable despite the composed task failing, and cluster count is decoded more accurately than touching-object count under the same probe setup. Read together with the behavioural comparison (Section 4.3), this motivates cluster counting as a candidate account without establishing that the model's readout uses this quantity or ignores the contact information (methodology in Appendix A.5.2; full numbers in Appendix A.10).

This is not the general case, though. Applying the same methodology to `radial-symmetry`'s and `mirror-symmetry`'s high-error-overlap constituents – with position, shape, and color independently randomized per probe trial, so that no probe can recover a relational target from a fixed canonical position alone – finds the opposite signature. Per-object exact-tolerance position, shape,

and color match are each decodable at or below their own majority-class base rate, at every depth we probe. For both rules, there is no evidence of linear encoding above chance at all (Appendix A.10).

4.5 Robustness

A descriptive check of whether this pattern tracks task structure – rules requiring a newly constructed relation collapse somewhat more than rules that only read an existing attribute – is in Appendix A.5.1.

Robustness checks, bundled. Additional experiments examine sensitivity to perception, architecture, training exposure, and intervention choice. *Perception*: a confound-isolation ablation (Appendix A.9) shows perception-free Oracle collapses the same way VQVAE and SlotAttn-Tr do, so pixel resolution is not doing the work. *Self-attention*: SlotAttn-RN swaps the shared Transformer for an attention-free Relation Network and reproduces the collapse almost cell-for-cell (Section 5). *Perception* and *self-attention together*: a pixel-space diffusion classifier, sharing neither a tokenized front end nor a relational Transformer with any other model in the roster, replicates the same shape-dependence collapse on the one rule where it learns well enough for the test to mean anything (Appendix A.16) – on the other eight, its own fit is too weak to be diagnostic, a limitation of that particular check rather than a finding about boundary collapse. *Training exposure*: including near-miss examples directly in training does not fix the collapse either – on `at-least-one-touching-pair`, even the oracle scores only 0.067 on a tight-contact pair, no better than a model whose training split saw no boundary examples at all (Appendix A.12). *Mechanism-level intervention*: the one attempt we made at this stronger test (Section 6) comes back a null result.

5 Related Work

Causal abstraction and mechanistic interpretability [11; 13; 12; 14] align neural representations with interpretable causal variables and validate the alignment via intervention; our profile targets a narrower question upstream of that – which named constituent a model's behaviour and representation actually support, before any causal-alignment claim. **Faithfulness:** Jacovi and Goldberg [33] argue faithfulness should be graded and operationalized, not asserted; our `CF`/`AM`/`E` triple is one concrete instantiation for relational rule-recovery.

Concept-level explanation and interpretability illusions. Standard feature attributions do not directly provide this constituent-level assessment: Gradient×Input, run against the same touch-boundary scene pair the oracle fails on, shifts substantially between the touching and near-miss constructions, but its output space – one relevance score per object, per unary feature – has no coordinate for "the model composed a touching predicate" versus "the model counts clusters, not objects"; LIME, SHAP, and Grad-CAM share this limitation by construction (Appendix A.19). Concept Relevance Propagation [40] answers *what* a decision represents, not only *where*, but a CRP concept is discovered post-hoc with no mechanism to score it against a named alternative. We run it directly, wherever it is actually computable in our roster, on SlotAttn-RN's attention-free Relation Network: its discovered concepts separate scenes by how much touching structure is present overall, but not either rule's matched True/False boundary pair (Appendix A.19). Bolukbasi et al. [34] and Makelov et al. [35] show an interpretation can appear to explain a model's behaviour while relying on a spuriously coincident pathway; our profile targets the same failure mode at the level of a recovered symbolic program rather than a subspace intervention.

Shortcut learning [37] documents that standard-distribution accuracy is compatible with a spurious decision rule in general; our contribution is narrower – a *correctly recovered symbolic program*, matching the true rule exactly, can still conceal which constituent's errors overlap with which shortcut. **Contrast sets [38] and counterfactual augmentation [39]** probe a model's local decision boundary with minimally edited examples, as we do; ours are additionally verified against an executable generating rule and mapped to a named constituent of an already-recovered program. **Kandinsky benchmarks** [5; 6; 7; 8] ask whether a model classifies a pattern correctly, learns an abstract rule, or generalizes to new compositions; we add whether its constituents are supported at the program's own decision boundary.

Disentangled representation learning. SlotAttn-Tr's Slot Attention encoder [1] targets the same broad goal disentanglement metrics evaluate, but those metrics ask only whether a factor is decodable,

not whether the model's decision uses it, and are undefined for a relational factor shared between two objects: the oracle decodes `touches(o)` at 94.5% while its own decision head fails at 19.6% (Section 4.4). **Representational universality:** the platonic representation hypothesis [36] argues capable models converge toward a shared representation; we measure a related but distinct quantity – whether independently recovered symbolic programs agree – and find they diverge even where behavioural agreement across VQVAE, SlotAttn-Tr, and Oracle is high (Appendix A.7). Recovered-program divergence is not itself a test of representational convergence.

Among the approaches compared above, none combines a per-constituent verdict, a boundary test, and a joint assessment of behavioural fidelity and representational decodability against verified ground truth (full comparison table in Appendix A.1).

6 Limitations and Conclusion

The method's scope is behavioural and representational throughout (Section 3); we did attempt the stronger, mechanism-level version of one test anyway, intervention commutativity under single-head ablation (Appendix A.11), which gives a negative result on the one rule/model pair where the question can even be posed – a limit on what single-head ablation can localize, not evidence against load-bearing structure existing elsewhere in the network. Our labeling thresholds lean conservative rather than generous: on `touching-count-matches-red-left`, `H_cluster` matches 100% of the oracle's (few) region-constituent errors, yet we still label that constituent *unexplained*, because its own `CF` (0.750) does not cross the threshold a cell needs before a high-error-overlap label applies. The rule library itself is small at the L3 threshold-Gestalt/symmetry tier, and every boundary-adversarial scene is hand-built against a known generating rule – what makes verification possible, and also why the paper does not show that such cases can be found automatically, at scale, from a recovered program alone. Two further checks address specific possible artifacts, not artifacts in general: regenerating the selected replication tasks from scratch and retraining every model on corrected geometry flips none of 126 directly comparable cells between collapsed and near-ceiling accuracy (Appendix A.15), ruling out the earlier scene-generator geometry bug as the source of the boundary-collapse pattern; and scoring an identified alternative against a chance baseline calibrated to its own error rate rather than a flat 50% floor, the oracle's holdout match still sits significantly above it while the two image-based models' sits significantly below (Appendix A.18), ruling out the choice of null as the source of that result.

Recovering a program that reproduces a model's decisions on the standard distribution answers only part of the question this paper asks. On Kandinsky Patterns, such a program can match the generating rule exactly and still fail to predict the model at boundary cases built to test the constituents it is made of; where it fails, an explicit alternative sometimes accounts for the discrepancy better, but only a held-out test distinguishes a real predictive improvement from a description that fits the sample it was read off. For the perception-free model, cluster counting is such an improvement, raising held-out agreement from 45.7% to 61.0%; for the two image-based models, the same alternative does not generalize. Per-object contact remains linearly decodable from the perception-free model's representation despite the model failing the composed task, adding a third, independent axis neither the boundary tests nor the alternative-hypothesis comparison could supply alone.

These conclusions are behavioural and representational throughout: a held-out-validated alternative is evidence it predicts a model's decisions better than the recovered rule, not evidence the model implements that computation, and decodability despite task failure does not establish whether the model uses that information. What a recovered rule explains about a model is accordingly not fixed by how well it reproduces the model's decisions on ordinary data – it is the outcome of auditing that program's constituents at their own boundaries, comparing named alternatives against held-out predictive performance, and probing what the representation makes available. The method's contribution is this audit: a way of reporting, cell by cell, where a recovered hypothesis holds, where an alternative improves it, and where the evidence available from behaviour and representation runs out, rather than collapsing all three into one number.

REPRODUCIBILITY STATEMENT

The relational DSL grammar, program-synthesis search procedure, every boundary-adversarial scene construction (verified against each rule's own executable evaluator), the five named alternative-hypothesis definitions, the linear-probe architecture and training procedure used for $\texttt{E}(c, \text{oracle})$, model architectures, training hyperparameters, and per-seed logs underlying every number in Section 4 are documented in the appendix and the anonymized code archive submitted as supplementary material (README with exact reproduction commands). `ground_truth_fidelity` and `decision_fidelity` are defined in Section 3; `CF`, `AM`, and `E` and their labeling thresholds are defined in Section 3. All `CF`/`AM`/`E` values are computed from saved evaluation logs or reproducible re-scoring of frozen checkpoints – no model is retrained to produce them; the geometry-corrected replication (Appendix A.15) is the one exception in this paper, a separately labeled robustness check whose method is retraining from scratch, not a source of any `CF`/`AM`/`E` number. SlotAttn-Tr's stochastic slot initialization is seeded explicitly in every script we release, and the seed used is stated alongside every SlotAttn-Tr number that depends on it.

ETHICS STATEMENT

All experiments use synthetic Kandinsky Pattern scenes generated under our own control; no human subjects or personal data are used or released in this paper. We are not aware of a specific foreseeable harm from this work's methodology or synthetic test bed.

AI USE STATEMENT

An AI coding/writing assistant (Claude) was used under the authors' direction to help draft and revise this paper's text, to compute `CF`/`AM`/`E` and the explanation profiles by re-scoring and aggregating already-existing, previously logged model predictions and checkpoints (no model in this paper was trained or retrained by the assistant), and to check internal numerical and cross-reference consistency against the authors' own experiment logs. All experimental design, code producing the underlying results, and final content were reviewed and approved by the authors.

REFERENCES

[1] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[2] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[3] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[4] A. Goldhirsch, E. P. Winer, A. S. Coates, R. D. Gelber, M. Piccart-Gebhart, B. Thürlimann, H.-J. Senn, and Panel members. Personalizing the treatment of women with early breast cancer: highlights of the St Gallen International Expert Consensus on the Primary Therapy of Early Breast Cancer 2013. *Annals of Oncology*, 24(9):2206–2223, 2013. DOI: 10.1093/annonc/mdt303.

[5] H. Müller and A. Holzinger. Kandinsky Patterns. *Artificial Intelligence*, 300:103546, 2021. DOI: 10.1016/j.artint.2021.103546.

[6] L. S. Lorello, M. Lippi, and S. Melacci. The KANDY benchmark: incremental neuro-symbolic learning and reasoning with Kandinsky patterns. *Machine Learning*, 114:161, 2025. DOI: 10.1007/s10994-025-06798-x.

[7] H. Shindo, V. Pfanschilling, D. S. Dhami, and K. Kersting. Learning differentiable logic programs for abstract visual reasoning. *Machine Learning*, 113:8533–8584, 2024. DOI: 10.1007/s10994-024-06610-2.

[8] A. Zerroug, M. Vaishnav, J. Colin, S. Musslick, and T. Serre. A benchmark for compositional visual reasoning. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. DOI: 10.52202/068431-2159.

[9] N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt. Progress measures for grokking via mechanistic interpretability. *International Conference on Learning Representations (ICLR)*, 2023.

[10] A. Daniely and E. Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[11] A. Geiger, H. Lu, T. Icard, and C. Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

[12] A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. DOI: 10.52202/075280-0719.

[13] A. Geiger, Z. Wu, H. Lu, J. Rozner, E. Kreiss, T. Icard, N. Goodman, and C. Potts. Inducing causal structure for interpretable neural networks. *International Conference on Machine Learning (ICML)*, PMLR 162:7324–7338, 2022.

[14] R. Lorenz and S. Tull. Causal and compositional abstraction. *arXiv:2602.16612*, 2026. DOI: 10.48550/arXiv.2602.16612.

[15] S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller. Sparse feature circuits: discovering and editing interpretable causal graphs in language models. *International Conference on Learning Representations (ICLR)*, 2025.

[16] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. *International Conference on Machine Learning (ICML)*, PMLR 119:5338–5348, 2020.

[17] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. GNNExplainer: generating explanations for graph neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[18] P. Veličković and C. Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021. DOI: 10.1016/j.patter.2021.100273.

[19] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. DOI: 10.1109/JPROC.2021.3058954.

[20] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, and J. B. Tenenbaum. DreamCoder: bootstrapping inductive program synthesis with wake-sleep library learning. *Programming Language Design and Implementation (PLDI)*, 835–850, 2021. DOI: 10.1145/3453483.3454080.

[21] T. Griebel, A. Archit, and C. Pape. Segment anything for histopathology. *Medical Imaging with Deep Learning (MIDL)*, PMLR 301:515–546, 2026.

[22] R. N. Shepard, C. I. Hovland, and H. M. Jenkins. Learning and memorization of classifications. *Psychological Monographs: General and Applied*, 75(13):1–42, 1961. DOI: 10.1037/h0093825.

[23] J. Feldman. Minimization of Boolean complexity in human concept learning. *Nature*, 407(6804):630–633, 2000. DOI: 10.1038/35036586.

[24] E. L. J. Leeuwenberg. Quantitative specification of information in sequential patterns. *Psychological Review*, 76(2):216–220, 1969. DOI: 10.1037/h0027285.

[25] E. L. J. Leeuwenberg. A perceptual coding language for visual and auditory patterns. *American Journal of Psychology*, 84(3):307–349, 1971. DOI: 10.2307/1420464.

[26] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. DOI: 10.1145/293347.293351.

[27] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[28] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. DOI: 10.1007/BF01744431.

[29] J. Håstad. Almost optimal lower bounds for small depth circuits. *Symposium on Theory of Computing (STOC)*, 6–20, 1986. DOI: 10.1145/12130.12132.

[30] W. Merrill and A. Sabharwal. The expressive power of transformers with chain of thought. *International Conference on Learning Representations (ICLR)*, 2024.

[31] S. W. McGrath and J. Russin. Multiple realizability and the rise of deep learning. *arXiv:2405.13231*, 2024. DOI: 10.48550/arXiv.2405.13231.

[32] J. Sanghavi. From edges to depth: probing the spatial hierarchy in Vision Transformers. *arXiv:2604.23452*, 2026. DOI: 10.48550/arXiv.2604.23452.

[33] A. Jacovi and Y. Goldberg. Towards faithfully interpretable NLP systems: how should we define and evaluate faithfulness? *Annual Meeting of the Association for Computational Linguistics (ACL)*, 4198–4205, 2020. DOI: 10.18653/v1/2020.acl-main.386.

[34] T. Bolukbasi, A. Pearce, A. Yuan, A. Coenen, E. Reif, F. Viégas, and M. Wattenberg. An interpretability illusion for BERT. *arXiv:2104.07143*, 2021. DOI: 10.48550/arXiv.2104.07143.

[35] A. Makelov, G. Lange, A. Geiger, and N. Nanda. Is this the subspace you are looking for? An interpretability illusion for subspace activation patching. *International Conference on Learning Representations (ICLR)*, 2024.

[36] M. Huh, B. Cheung, T. Wang, and P. Isola. Position: the platonic representation hypothesis. *International Conference on Machine Learning (ICML)*, PMLR 235:20617–20642, 2024. `https://proceedings.mlr.press/v235/huh24a.html`.

[37] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. DOI: 10.1038/s42256-020-00257-z.

[38] M. Gardner, Y. Artzi, V. Basmov, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala, N. Gupta, H. Hajishirzi, G. Ilharco, D. Khashabi, K. Lin, J. Liu, N. F. Liu, P. Mulcaire, Q. Ning, S. Singh, N. A. Smith, S. Subramanian, R. Tsarfaty, E. Wallace, A. Zhang, and B. Zhou. Evaluating models' local decision boundaries via contrast sets. *Findings of the Association for Computational Linguistics: EMNLP*, 1307–1323, 2020. DOI: 10.18653/v1/2020.findings-emnlp.117.

[39] D. Kaushik, E. Hovy, and Z. C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *International Conference on Learning Representations (ICLR)*, 2020.

[40] R. Achtibat, M. Dreyer, I. Eisenbraun, S. Bosse, T. Wiegand, W. Samek, and S. Lapuschkin. From attribution maps to human-understandable explanations through Concept Relevance Propagation. *Nature Machine Intelligence*, 5:1006–1019, 2023. DOI: 10.1038/s42256-023-00711-8.

A Appendix

A.1 Related Work: Full Comparison Table

Section 5 argues each column below; this table gives the full comparison at a glance.

Approach	GT?	Per-constituent?	Tests boundary?	Encoding ≠ use?
Causal abstraction [11; 13; 12]	–	–	–	–
CRP [40]	–	–	–	partial (post-hoc, unnamed)
Contrast sets [38; 39]	–	–	yes	–
Attribution methods	–	–	–	–
Disentanglement [1]	yes (labels)	– (1-obj. only)	–	– (decode only)
Ours	yes	**yes**	**yes**	**yes**

Among the approaches compared above, none combines a per-constituent verdict, a boundary test, and a joint assessment of behavioural fidelity and representational decodability against verified ground truth. "Attribution methods": Grad-CAM, SHAP, LIME.

A.2 A Worked Example: Two Kandinsky Scenes



Figure 2: Two scenes under `touching-count-matches-red-left`, the rule this paper's flagship results (Section 4.2) are built around: the number of objects touching some other object must equal twice the number of red objects in the left third of the image. *Left (label TRUE)*: the blue triangle and circle touch (2 touching objects; the two red objects at bottom are separate from each other and from everything else), and 1 red object sits in the left region ($2 = 2 \times 1$). *Right (label FALSE)*: no two objects touch, while 2 red objects sit in the left region ($0 \neq 2 \times 2$). Both scenes verified against `physics.touches()` and `rule.evaluate()` on the exact object placements shown, not assumed from the layout. Deciding this rule requires relating objects to each other (contact) and to a region (left third), not reading off any single object's own attributes – the $c_{\text{construct}} = 1$ property (Section 2) made concrete.

A.3 Relational DSL Grammar

Every rule in this paper's library is an abstract syntax tree over a fixed grammar: unary attribute predicates `Atom1` (color, shape, region membership); binary relational predicates `Atom2` (e.g. `touches`, `left_of`, `above`); boolean connectives `And`, `Or`, `Not`; quantifiers `Exists` and `Forall` over the scene's object set; a counting operator `CountOf` (cardinality of the set of objects satisfying a sub-formula); and an arithmetic comparison operator `CountCompare` combining two `CountOf` terms (optionally scaled by an integer `Const` via `Mul`) under $=, >, <, \geq$, or $\leq$. Program synthesis (Step 1, Section 3) enumerates candidates in this grammar, scored against a model's own decisions on held-out scenes, and returns the smallest candidate reaching a fidelity threshold; ties are broken toward fewer AST nodes.

A.4 Relation Types and Semantic Typing

The relational vocabulary S_R's typed relations draw on covers: adjacency, containment, overlap, boundary crossing, grouping, identity, comparison, counting, branching, logical implication, symmetry, closure, *quantification* ($\exists/\forall$ over a set of entities, with negation), and *threshold comparison* (a

count against a fixed constant, as opposed to **comparison** in the narrower sense of one count against another). None of these types are directly readable off a model's raw decisions or representation; assigning one to a recovered constituent is what program synthesis and the constituent-level audit (Section 3) do.

This vocabulary is domain-general (reusable across Kandinsky or any other domain the same kind of program synthesis is applied to) and is what Section 3's definition of S_R's typed relations draws on.

A.5 The Five-Axis Complexity Profile

Section 2 summarizes each rule by $c_{\text{construct}}$ alone, the one axis Section A.5.1 uses; in full, each rule R has a five-axis profile

$$\mathcal{C}(R) = \big(c_{\text{arity}}(R),\, c_{\text{quant}}(R),\, c_{\text{agg}}(R),\, c_{\text{construct}}(R),\, c_{\text{alg}}(R)\big),$$

read off R's minimal AST: c_{arity}, the largest number of jointly-referenced entities any single relation or operator needs; c_{quant}, quantifier depth and reach; c_{agg}, aggregation/arithmetic cost; $c_{\text{construct}}$, whether the rule requires a newly constructed relational predicate not already given as an attribute (0 unless the rule must derive something like contact or a symmetric partner from raw coordinates rather than read an existing field); and c_{alg}, an SQ/algebraic-hardness degree we do not use in this paper. $c_{\text{construct}}$ is the axis Section A.5.1's descriptive comparison is built on, because it is the one axis directly tied to the boundary-adversarial audit's own logic (Section 3): a rule with $c_{\text{construct}} = 1$ has a constituent – the constructed predicate itself – that a coarse proxy (an approximate distance threshold, an approximate symmetry check) can satisfy everywhere a standard test distribution samples without computing the exact predicate, which is the failure mode Section 4.2 documents. The other four axes and the c_{alg}/SQ-hardness argument are not used by any other claim in this paper's main text.

The rule library also groups rules into five descriptive tiers, L0–L4, reflecting the AST shape of each rule's minimal program:

Level	Pattern observed for rules at this level	Example
L0	A single existential or count-vs-constant check over given attributes; no boolean combination of clauses, no counting beyond one threshold.	`majority-red-of-four`: count(red) ≥ 3 of 4 objects.
L1	A boolean combination (conjunction, implication, or universal quantification) of two or more clauses over given attributes, including a given (not constructed) binary relation; no counting.	`left-red-implies-right-blue`: a red object in the left region implies a blue object in the right region.
L2	Counting or arithmetic beyond a single count-vs-constant check: a count-vs-constant wrapped in a conditional, or a count compared against another count.	`yellow-count-mod-four-equals-blue-count`: count(yellow) mod 4 equals count(blue).
L3	Requires constructing a new relational predicate from raw coordinates rather than reading a given attribute ($c_{\text{construct}} = 1$).	`touching-count-matches-red-left`: the touching-object count.
L4	A parity/XOR-style combination of independent unary indicators, with no continuous decision boundary.	`parity-five-indicators`: an odd number of 5 (shape, color) indicators occur.

This grouping does not order rules by proven learning difficulty, required network depth, or a proven complexity lower bound; c_{alg}, the axis that would carry a formal hardness argument, is defined above but not used by any claim in this paper. The one part of the grouping the paper's own results depend on is L3's link to $c_{\text{construct}} = 1$, verified directly against each rule's AST (Section A.5.1). For reference, the L0–L4 tier and $c_{\text{construct}}$ for each rule in this paper's library:

Rule	Lvl	constr.	Rule	Lvl	constr.
left-red-implies-right-blue	L1	0	at-least-one-touching-pair	L3	1
red-count-triples-yellow-triangles	L2	0	touching-count-matches-red-left	L3	1
yellow-count-mod-four-equals-blue-count	L2	0			
parity-five-indicators	L4	0			
radial-symmetry	L3	1			
mirror-symmetry	L3	1			

A.5.1 Does task structure predict where fit misleads?

Takeaway: rules that require deriving a new geometric relation are, on average, harder to verify at their own boundary than rules that just read an existing attribute.

We can ask, descriptively, whether this pattern tracks task structure at all. Pooling every constituent's `CF` per rule into a single "boundary accuracy" number, the four rules with $c_{\text{construct}} = 0$ average 0.532; the four rules with $c_{\text{construct}} = 1$ average lower, 0.433, driven largely by `radial-symmetry` and `mirror-symmetry` (0.268, 0.264) – though `at-least-one-touching-pair`, at 0.641, is a clear counterexample within the same group. The direction at least fits why $c_{\text{construct}} = 1$ is the axis we built this comparison on in the first place: such a rule has a constituent that a coarse proxy can satisfy everywhere the standard distribution samples, without ever computing the exact predicate it is supposed to stand in for (Section 4.2), while a predicate that only reads an existing attribute offers no such proxy to hide behind. The overlap between the two groups is real, though, so we treat this as no more than a tendency worth testing on a larger rule library, not an established effect.

A.5.2 Model-side: where and how a trained model represents structure

Whether a specific trained model actually recovers a rule's structure at a given representation depth is a model-side question, for which we describe three complementary measurements.

Ablation-based causal localization (implemented). Individual attention heads of a trained relational Transformer are zero-ablated one at a time (Phase 5, the appendix), and the resulting change in decision agreement is measured against a fixed threshold. This is a direct, causal (not merely correlational) intervention on M_x itself: it answers *which components are load-bearing*, but not yet *what relational computation* they perform, nor *when in training or at what depth* that computation becomes available — the gap the next two measurements are designed to close.

Per-layer linear probing (implemented). A classic diagnostic: a linear classifier is trained on a model's frozen hidden activations at each of three depths (before any self-attention layer, after layer 1, after the final layer) to predict the true label, tracking probe accuracy as a function of depth. This tests, per model, the one part of the L0/L4 tiers (Section A.5) that rests on an actual theorem rather than the tiers' own descriptive pattern: a conjunctive (AND/OR-like) concept is linearly separable and should show high probe accuracy already at an early layer, whereas parity of $k \geq 2$ bits is the textbook example of a concept no linear (single-threshold) function can separate at all [27] – a probe can only succeed here by relying on the network's own *non-linear* layers to have already recombined the bits into something linearly readable, so probe accuracy should stay near chance until whatever depth (if any) performs that recombination. (AC^0 circuit lower bounds [28; 29] are a different, asymptotic argument about growing-input-size circuit families and are not what license this particular fixed-k prediction.) Run across six rules spanning L0–L4 (the L0 anchor, `onered`, is drawn from the broader rule library specifically for this check, not one of this paper's eight testbed rules), only the two extremes land cleanly on this prediction: L0 is near-ceiling before any attention layer, and L4 stays flat at chance throughout, mirroring the data-side signature exactly. L1, L2, and both L3 rules do *not* show a clean rise with depth, for two identified reasons that are themselves methodological findings: the depth-0 readout is already a mean pool over tokens, so it is an aggregation baseline rather than a no-aggregation baseline, and at the coarser tokenization used for most of these checkpoints a single token's receptive field can span multiple nearby objects, letting proximity-based relations leak in through convolution before any attention layer runs. A rerun on a finer-grid checkpoint partially confirms the second explanation: the depth-0 accuracy for a touch-based L3 rule drops and a depth-0-to-1 rise appears that the coarser checkpoint did not show. Repeating this rerun across 3 seeds (reusing the checkpoints already produced by the confound-isolation ablation above, at no additional training cost) confirms the depth-0-to-1 rise is not a single-checkpoint artifact: it appears in all 3 individual seeds for both a threshold-Gestalt rule ($0.829 \to 0.918 \to 0.936$) and a smooth-Gestalt rule ($0.753 \to 0.853 \to 0.860$). Full tables and discussion are in the appendix.

Diff-of-means direction extraction from counterfactual pairs (implemented). The near-miss and minimal-pair construction already used to build training data (the appendix) is equally well suited to a different purpose: for a matched pair of scenes differing in exactly one relevant factor, the difference between their final-layer representations isolates a candidate direction encoding that specific factor, evaluated on held-out pairs via a midpoint-threshold projection accuracy. Run against the originally available exact boundary-case pairs (3–8 per attribute), the method gave no clean signal, near-identical across 3 seeds, consistent with either a too-small pool or an absent linear direction – the two explanations indistinguishable at that sample size. Regenerating a larger boundary-case pool

(20 pairs per attribute, up from 3–8) and repeating the measurement across the same 3 seeds resolves the ambiguity partially: `position` now shows real, seed-varying signal (projection accuracy 0.525–0.775 across seeds, mean 0.633 ± 0.128) rather than the identical chance-level number every seed gave at the smaller pool, while `shape` remains at exactly 0.500 in all 3 seeds even with $4\times$ more data, and `colour`/`rotation`/`size` stay close to chance (0.517–0.533 mean). This pattern – one attribute's signal strengthening substantially with more data while others stay flat – is what the sample-size account predicts and the null-direction account does not, at least for `position`; `shape` landing at exactly chance across every seed at both pool sizes remains equally consistent with either account for that specific attribute.

A.6 Category-to-Constituent Mapping and Full Profile Table

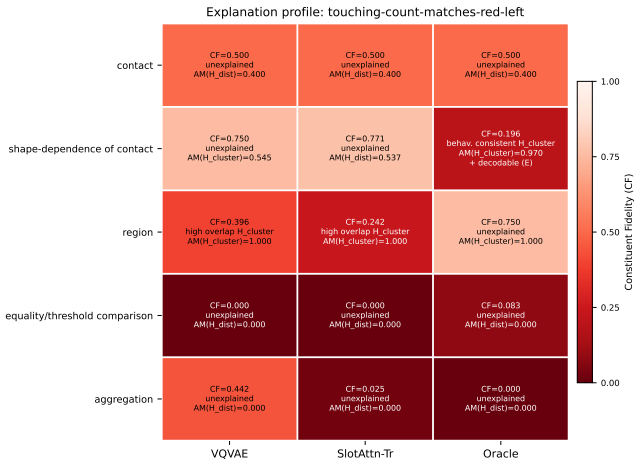


Figure 3: Explanation profile for `touching-count-matches-red-left`: constituent fidelity (color and CF value) and, where a labeling threshold is crossed, the best-attributed alternative. *High overlap H_cluster* fires for region (VQVAE/SlotAttn-Tr) – raw error overlap only, not holdout-tested, and contradicted by the holdout test for these two models (Appendix A.18); shape-dependence of contact (Oracle) is the one cell additionally holdout-validated, so it is marked *behav. consistent H_cluster* and separately + *decodable (E)*; every other cell is *unexplained*.

Every named alternative is one of five AST-edit types. Section 3 states this in the main text; the table below gives the full mapping, all 19 alternatives used in this paper against the recovered program's own AST.

Edit type	Alternatives	Rules
Drop a conjunct	`H_allred`, `H_anyblue`, `H_anyred`, `H_allyellow`, `H_xor4`, `H_anyshape`	touching-count-matches-red-left, left-red-implies-right-blue (x2), red-count-triples-yellow-triangles, parity-five-indicators, red-triangle-proportion-ge-20pct
Shift a threshold constant	`H_cutoff`, `H_shift_ante`, `H_shift_conseq`, `H_gel1`	touching-count-matches-red-left, left-red-implies-right-blue (x2), red-triangle-proportion-ge-20pct
Widen a tolerance parameter	`H_approx` (x2), `H_loose_tol`	touching-count-matches-red-left, red-count-triples-yellow-triangles, radial-/mirror-symmetry
Change a comparison operator	`H_ge2`, `H_eq4`, `H_noWrap`	majority-red-of-four (x2), yellow-count-mod-four-equals-blue-count
Change an aggregation function	`H_cluster`, `H_ignore_type`, `H_majority`, `H_dist` (x3)	touching-count-matches-red-left, radial-/mirror-symmetry, parity-five-indicators, at-least-one-touching-pair

`H_dist` (shape-blind distance contact) is reused verbatim across three rules with a touching-based contact constituent; `H_approx` is independently instantiated per rule (different arithmetic each time) but is the same edit type. This is a fixed, 5-item operator set applied uniformly to whichever rule's own AST is at hand, including for `parity-five-indicators`: read through this taxonomy, dropping one of its five XOR conjuncts (`H_xor4`) and substituting a majority vote for parity (`H_majority`) are both legitimate candidates, tested in Appendix A.14.

Every boundary-adversarial category is mapped to exactly one constituent (contact, shape-dependence of contact, region, equality/threshold comparison, aggregation); where a rule has no touching relation at all (`left-red-implies-right-blue`, the two arithmetic/count rules, the two symmetry rules), "shape-dependence of contact" is read as the general case it instances – does the model's relational predicate depend correctly on shape/color identity, not just position or count – an explicit extension of the taxonomy from the touching-based rules it was worded around, stated here rather than silently assumed.

Where a constituent's category set mixes a sanity baseline (an easy, non-adversarial construction) with a targeted boundary category, `CF` below is computed on the boundary category alone; the baseline is instead used as a validity gate (the model must score ≥ 0.95 on it, otherwise that model's boundary result for the constituent is not treated as diagnostic) and is reported separately, not pooled into `CF`. This applies to `touching-count-matches-red-left`'s equality/threshold comparison constituent (baseline `arithmetic_baseline_true`, ≥ 0.95 for all three models). It does *not* apply to contact: `touch_boundary_true` is itself the tightest-possible still-touching construction, not an easy baseline (Appendix A.8), so contact's pooled 0.500 is a two-sided boundary result, not a baseline/boundary mixture – pooling a true baseline with a fully collapsed boundary category is what mechanically drives `CF` toward 0.5 and is what the gate above is designed to prevent.

Rule	Constituent	Model	CF	CF 95% CI	CF n	Label
touching-count-matches-red-left	contact	VQVAE/SlotAttn-Tr/Oracle	0.500	[0.412,0.588]	120	unexplained
touching-count-matches-red-left	region	VQVAE	0.396	[0.336,0.459]	240	high error overlap with H_cluster (does not hold up in holdout test, App. A.18)
touching-count-matches-red-left	region	SlotAttn-Tr	0.242	[0.192,0.300]	240	high error overlap with H_cluster (does not hold up in holdout test, App. A.18)
touching-count-matches-red-left	region	Oracle	0.750	[0.692,0.801]	240	unexplained
touching-count-matches-red-left	equality/threshold comparison (boundary-only)	VQVAE/SlotAttn-Tr	0.000	[0.000,0.060]	60	unexplained
touching-count-matches-red-left	equality/threshold comparison (boundary-only)	Oracle	0.083	[0.036,0.181]	60	unexplained
touching-count-matches-red-left	aggregation	VQVAE	0.442	[0.356,0.531]	120	unexplained
touching-count-matches-red-left	aggregation	SlotAttn-Tr	0.025	[0.009,0.071]	120	unexplained
touching-count-matches-red-left	aggregation	Oracle	0.000	[0.000,0.031]	120	unexplained
touching-count-matches-red-left	aggregation	SlotAttn-RN	0.042	[0.018,0.094]	120	unexplained
touching-count-matches-red-left	shape-dependence of contact	VQVAE	0.750	[0.712,0.785]	537	unexplained
touching-count-matches-red-left	shape-dependence of contact	SlotAttn-Tr	0.771	[0.734,0.804]	537	unexplained
touching-count-matches-red-left	shape-dependence of contact	Oracle	0.196	[0.164,0.231]	537	behaviourally consistent with H_cluster; linearly decodable despite task failure
left-red-implies-right-blue	region	VQVAE/SlotAttn-Tr/Oracle	0.45–0.52	–	360	unexplained
red-count-triples-yellow-triangles	aggregation	VQVAE/SlotAttn-Tr	0.34–0.37	–	60	high error overlap with H_allyellow
red-count-triples-yellow-triangles	aggregation	Oracle	0.967	[0.886,0.991]	60	unexplained (closest to passing the constituent boundary test of any cell)
red-count-triples-yellow-triangles	aggregation	SlotAttn-RN	0.400	[0.286,0.526]	60	unexplained (`H_allyellow` attribution 100%, but `CF` upper bound misses the high-error-overlap cutoff by a narrow margin)
red-count-triples-yellow-triangles	equality/threshold comparison	VQVAE/SlotAttn-Tr/Oracle	0.46–0.80	–	300	unexplained
yellow-count-mod-four-equals-blue-count	equality/threshold comparison	Oracle	0.729	[0.696,0.760]	720	unexplained
radial-symmetry	equality/threshold comparison	VQVAE/SlotAttn-Tr/Oracle	0.33–0.37	–	180	high error overlap with H_loose_iol
radial-symmetry	shape-dependence of contact	VQVAE/SlotAttn-Tr/Oracle	0.01–0.33	–	120	high error overlap with H_ignore_type
radial-symmetry	shape-dependence of contact	SlotAttn-RN	0.142	[0.090,0.215]	120	high error overlap with H_ignore_type
mirror-symmetry	equality/threshold comparison	VQVAE/SlotAttn-Tr/Oracle	0.33–0.36	–	180	unexplained
mirror-symmetry	shape-dependence of contact	VQVAE/SlotAttn-Tr/Oracle	0.04–0.33	–	120	high error overlap with H_ignore_type
mirror-symmetry	shape-dependence of contact	SlotAttn-RN	0.075	[0.040,0.136]	120	high error overlap with H_ignore_type
parity-five-indicators	aggregation	VQVAE/SlotAttn-Tr/Oracle	0.48–0.51	–	1200	unexplained
at-least-one-touching-pair	contact	VQVAE/SlotAttn-Tr/Oracle	0.61–0.66	–	180	unexplained
majority-red-of-four	equality/threshold comparison	VQVAE	0.808	[0.729,0.869]	120	unexplained
majority-red-of-four	equality/threshold comparison	SlotAttn-Tr	0.633	[0.544,0.714]	120	unexplained
majority-red-of-four	equality/threshold comparison	Oracle	**1.000**	[0.969,1.000]	120	**passes the constituent boundary test**
red-triangle-proportion-ge-20pct	equality/threshold comparison	VQVAE	0.483	[0.396,0.572]	120	unexplained
red-triangle-proportion-ge-20pct	equality/threshold comparison	SlotAttn-Tr	0.517	[0.428,0.604]	120	unexplained
red-triangle-proportion-ge-20pct	equality/threshold comparison	Oracle	0.658	[0.570,0.737]	120	unexplained

Full 72-row table (the original 8 rules plus two added separately, `majority-red-of-four` – Section 4.1 – and `red-triangle-proportion-ge-20pct` – Section 4.1, the Ki-67-motivated rule, no SlotAttn-RN checkpoint trained on this 60-triangle/12-threshold construction, the one these published `CF`/`AM` numbers come from (a separate, rescaled 120-triangle/24-threshold construction, used only for the implicit-threshold check in Appendix A.15, does have its own SlotAttn-RN checkpoints) – every constituent, all four models where trained (VQVAE, SlotAttn-Tr, Oracle, and SlotAttn-RN, the attention-free Relation Network variant added as a robustness check – Appendix A.19), with per-category source columns) in `profiles.csv`, released with the anonymized code archive; the labeling thresholds (passes the constituent boundary test, high error overlap with a or, where separately holdout-validated, behaviourally consistent with a, behaviourally inconsistent/no single alternative identified, unexplained – plus the independent linearly-decodable-despite-task-failure flag) are defined in Section 3 and applied identically to every row – no threshold was adjusted after seeing a result. `red-triangle-proportion-ge-20pct`'s best-attributed alternative, `H_anyshape`, is exactly indistinguishable from `H_ge11` (loosen the threshold by one instead of dropping the shape conjunct) on every scored scene: with 30 uniformly-colored decoy circles, at least one is red in nearly every scene, so a shape-blind miscount and an off-by-one threshold error coincide almost perfectly by construction (pooled AM 0.063/0.089/0.878 for VQVAE/SlotAttn-Tr/Oracle, both alternatives tied exactly). The pooled `CF` above splits into two boundary categories with the opposite sign per architecture: VQVAE/SlotAttn-Tr collapse on the true-boundary category (0.033/0.150) while scoring near ceiling on the false one (0.933/0.917); Oracle does the reverse (0.917 true, 0.400 false) – the same 20% cutoff, opposite miscalibration direction by architecture.

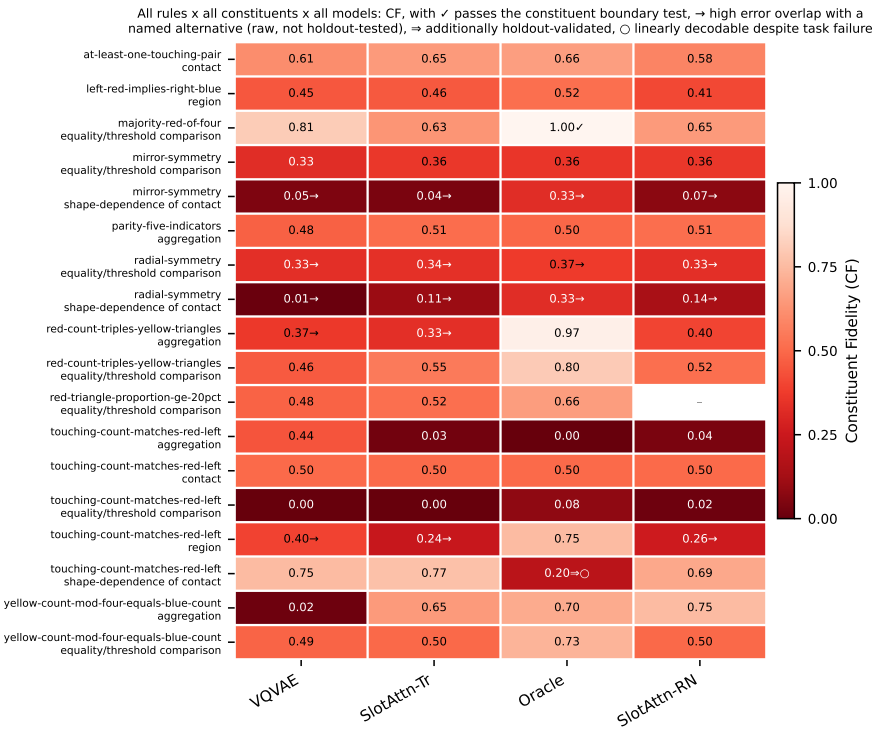


Figure 4: The same table as a heatmap: every (rule, constituent) row (18 in total) by model (4) cell in `profiles.csv` at once. Near-uniform collapse, with the two cells that pass the constituent boundary test or come closest to it (both Oracle) as the visible exceptions; ⇒ marks the one cell additionally holdout-validated (Section 4.3), → raw error overlap only – referenced from Section 4 as the overview the per-rule tables and prose in Section 4 then unpack.

A.7 BEHAVIOURAL STRUCTURE RECOVERY ACROSS THE MODEL ROSTER

Recovered structure is evaluated behaviourally ($f(x) \rightarrow S$, via enumerate-and-score program synthesis against each model's own decisions, no ground-truth access) across a 20-rule library (a broader survey than the 8-rule boundary-adversarial subset the rest of this paper centers on) and VQVAE, SlotAttn-Tr, and Oracle (the perception-free upper bound); full search-space and selection detail, including why this figure is closer to in-sample fit within a small, fixed template space than to an independent generalization estimate, is in the appendix. Mean behavioural fit accuracy, 3 seeds per model per rule: VQVAE 0.800, SlotAttn-Tr 0.738, Oracle 0.940. VQVAE beats SlotAttn-Tr on 17 of 20 comparable rules, net evidence that discrete patch tokens are not disadvantaged relative to object-centric slots for this recovery task (H15.2); the clearest exception is `radial-symmetry` (SlotAttn-Tr 0.960 vs. VQVAE 0.777, the strongest object-centric advantage in the library) – `touching-count-matches-red-left`'s own near-tie here (0.923 vs. 0.904) is the rule Section A.8 tests for boundary robustness.

Cross-model structural convergence – comparing VQVAE, SlotAttn-Tr, and Oracle's independently recovered ASTs against each other, not only against ground truth – is largely seed-fragile: only `touching-count-matches-red-left` converges robustly (tree edit distance 0 between every model pair, every seed, zero variance; Section A.8 tests whether that convergence reflects a boundary-robust computation at all). Every other rule shows a non-zero edit distance (5–14) even when raw decision agreement is high (Figure 5) – high behavioural agreement does not imply structural invariance. Full per-rule breakdown and the execution-agreement cross-check are in the appendix.

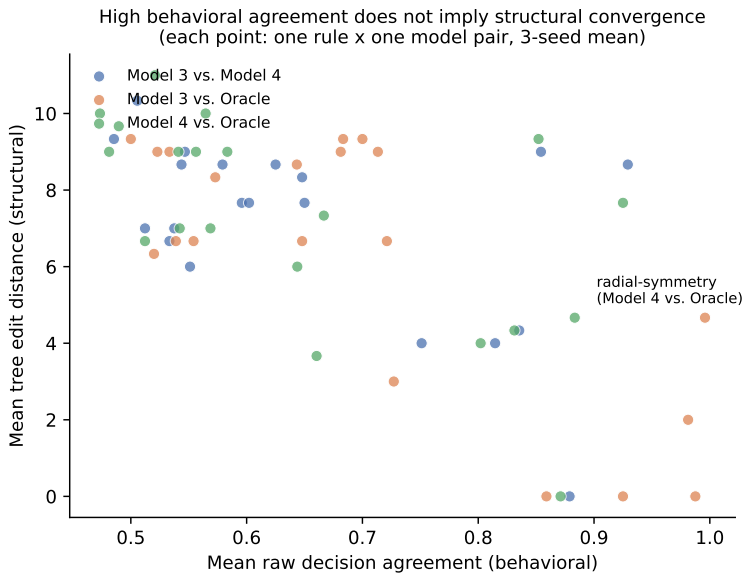


Figure 5: Behavioral agreement (x-axis) vs. structural convergence (y-axis, lower is more convergent), one point per rule per model pair, 3-seed means. Points near the top-right – high behavioral agreement, high structural distance – are the H1/H5 falsification pattern: models that mostly agree on classification without having recovered the same structure.

A worked example of a partial recovery (correct touching/counting structure, wrong region predicate, dropped factor-of-two) is in the appendix, alongside the full realization-fidelity results: regenerating scenes from a recovered rule and checking the model's own classification on them exposes a gap invisible to static fit accuracy. Headline number: on `radial-symmetry`, SlotAttn-Tr's want_true realization fidelity is effectively undefined (generation_success_rate ≈ 0 in 2 of 3 seeds) and 0.000 in the seed where it can be measured, despite mean behavioural fit accuracy of 0.961; Oracle's want_true realization fidelity is 0.133 against 0.916 fit accuracy. This tests H4 against a behaviourally recovered S_R, not a candidate validated by intervention commutativity, and establishes that behavioural fit accuracy alone does not guarantee a candidate structure reproduces model behaviour on freshly generated instances of its own predicted class.

A.8 BOUNDARY-ADVERSARIAL STRESS TESTING: EXACT CONSTRUCTION

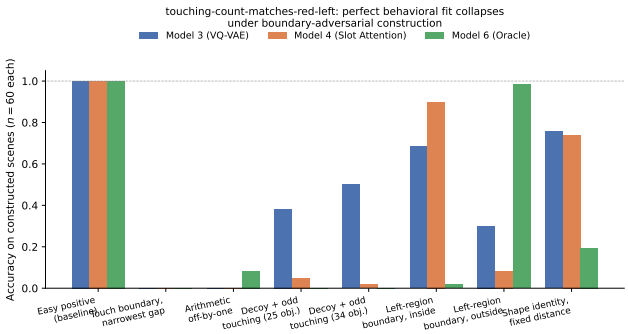


Figure 6: Standard-distribution fit is 1.0 for all three models; every constituent's boundary categories collapse, none uniformly – the same numbers as the table in Section 4.2.

The exact construction of each category referenced from Section 4.2 (each verified against `rule.evaluate()` before scoring, none used for training or selection): (i) *touch boundary* – the narrowest non-touching gap `physics.touches` itself can produce; (ii) *region boundary* – a red object placed 0.002 to either side of the `left_region` cutoff ($x = 1/3$), all else held fixed; (iii) *arithmetic off-by-one* – a correct positive scene plus one additional, otherwise ordinary red-left object (touching count unchanged, red-left count $+1$); (iv) *decoy odd-touching* – more than ten small red circles in `right_region` (must not be counted) plus a touching cluster of exactly 19 objects (odd, so never $2\times$ any integer red-left count), at 25 and 34 total objects; (v) *shape identity* – center-to-center distance and size held fixed while only shape identity is varied across all 9 (shape, shape) combinations, since `physics.touches`'s true contact distance depends on shape and rotation as well as size (settled contact distance ranges 0.056–0.162 across this rule's shape/size/rotation combinations, so at least one of the 9 combinations' ground-truth label differs from the rest by construction, every trial).

Stress category	VQVAE	SlotAttn-Tr	Oracle	SlotAttn-RN
Easy positive (sanity baseline, $n = 60$)	1.000	1.000	1.000	1.000
Touch boundary, narrowest non-touching gap ($n = 60$)	**0.000**	**0.000**	**0.000**	**0.000**
Arithmetic off-by-one (extra red-left object, $n = 60$)	**0.000**	**0.000**	0.083	**0.017**
Decoy reds + odd touching count, 25 objects ($n = 60$)	0.383	0.050	**0.000**	0.050
Decoy reds + odd touching count, 34 objects ($n = 60$)	0.500	0.017	**0.000**	0.033
Left-region boundary, just inside $x = 1/3$ ($n = 60$)	0.683	0.933	0.017	0.883
Left-region boundary, just outside $x = 1/3$ ($n = 60$)	0.300	0.083	0.983	0.100
Shape identity at fixed distance/size (n=537 pooled)	0.750	0.771	**0.196**	0.695

Figure 7 plots this table.

Discussion of this table (touch-boundary and arithmetic collapse, the model-specific region-threshold bias, and the shape-identity reversal) is in Section 4.2. One finding not in the main text: a follow-up check isolates the shape-blindness component from the other three. Retraining Oracle with one extra, precomputed per-object feature – whether that object touches something, the exact fact `physics.touches` already determines and the true rule sums over – fixes `shape_blindness` completely (1.000 on all 9 shape combinations, up from 0.067–0.400), while leaving `touch_boundary_false_near_miss`, `arithmetic_off_by_one`, and

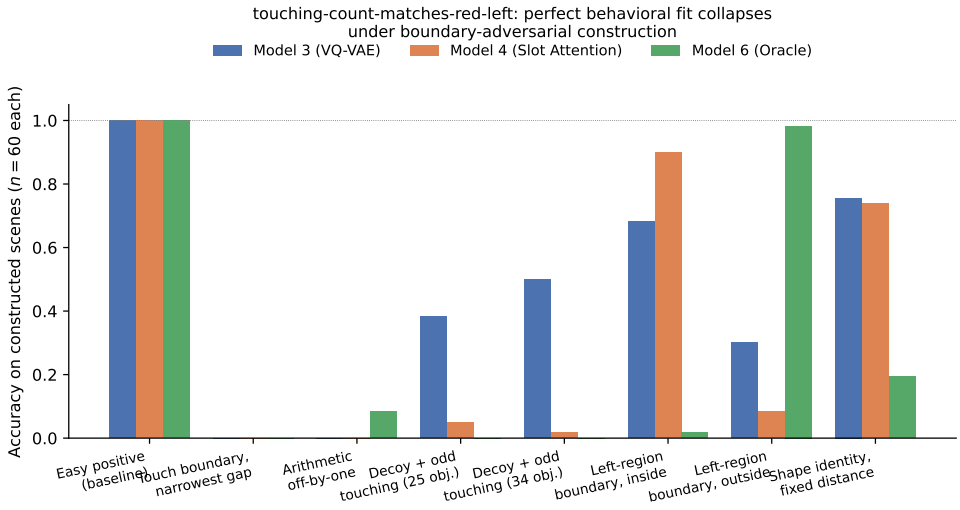


Figure 7: The table above, visualized: every model scores 1.0 on the easy baseline and collapses toward 0 on most adversarial categories, including Oracle (exact symbolic input, no perception stage). The rightmost category inverts the usual pattern: Oracle is the *worst* performer, not the best, when shape identity alone determines the label.

`decoy_odd_touching` unchanged at 0.000. Shape-derivation, exact-count arithmetic, and region-boundary calibration are three independent failure modes, not one masking the other two: removing the first leaves the other two completely untouched.

A.9 A CONFOUND-ISOLATION ABLATION FOR HIGH-RESOLUTION PERCEPTION

VQVAE's VQ-VAE tokenizer and the shared relational Transformer were extended to a finer 32×32 image-token grid (from the default 8×8) with a learned 2D positional embedding, motivated by touch-based rules needing fine-grained spatial reasoning. Isolating resolution, positional encoding, and training-data density as three independently varied factors (3 seeds per configuration; full methodology in the appendix) on `touching-count-matches-red-left`:

Configuration	Resolution	Pos. enc.	Data density	Mean val. acc. (std)
Baseline	8×8 grid	n/a	old	88.6% (3.4)
Full change	32×32 grid	on	new	**94.9%** (0.8)
No positional encoding	32×32 grid	off	new	87.1% (3.4)
Old data density	32×32 grid	on	old	94.2% (2.7)
New data density only	8×8 grid	n/a	new	88.4% (1.4)

Figure 8 plots these means.

Resolution and positional encoding together account for the gain (both configurations combining them reach ≈94–95%, well outside either configuration's own seed-to-seed noise); training-data density alone contributes nothing distinguishable from baseline; positional encoding is necessary for the resolution increase to pay off. With only 3 seeds per cell these are overlapping-std comparisons, not formal significance tests (a caveat and one learning-rate confound in the old-data-density row are detailed in the appendix). Applying the same recipe to `radial-symmetry` reaches 88.7% (±3.1), with a transient training instability in 1 of 3 seeds.

A.10 MODEL-SIDE STRUCTURAL PROBING

Layer-wise linear probing (frozen activations at three depths: before self-attention, after layer 1, after the final layer) against six rules spanning L0–L4 shows the predicted rise-with-depth pattern cleanly only at the extremes: L0 (`onered`) is near-ceiling before any attention layer (0.962→0.975);

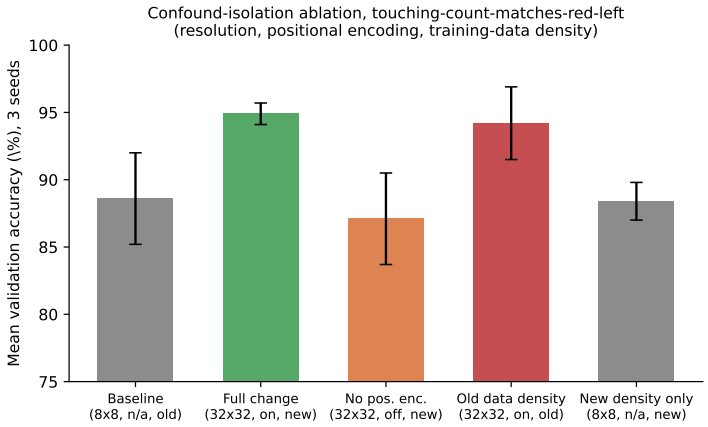


Figure 8: Mean validation accuracy (3 seeds, error bars = 1 std) for each configuration in the confound-isolation ablation. Resolution and positional encoding together (green) account for the gain over baseline (gray); removing positional encoding while keeping the finer grid (orange) does not improve on baseline; data density alone (gray, rightmost) is likewise indistinguishable from baseline.

L4 (`parity-five-indicators`) stays flat at chance throughout (0.513→0.523). At the high-resolution 32×32 grid (3 seeds), both L3 rules tested show a depth-0-to-1 rise in every individual seed (`touching-count-matches-red-left` $0.829 \rightarrow 0.918 \rightarrow 0.936$; `radial-symmetry` $0.753 \rightarrow 0.853 \rightarrow 0.860$) – a finer token grid changes not only how well but *how* (at what depth) the model computes the rule. Ablation-based causal localization (the mechanism-localization script, zero-ablating individual attention heads), run at 3 seeds each for `touching-count-matches-red-left` on VQVAE and Oracle, gives a direct causal result: it has a causally load-bearing head in *every* seed for both models, always in layer 1, though *which* head within that layer is seed-dependent (full per-seed head tables in the appendix).

Constituent probes for radial-symmetry / mirror-symmetry. Probe scenes for these two rules (`eval/probe_constituents_symmetry_unconfounded.py`) randomize the target pair's base position, position-match tolerance level, shape match, and color match independently per trial (2 exact anchor pairs plus 1 randomized target pair, matching each stress test's own scene shape): a fixed canonical target position would let a linear probe recover the label from raw coordinates with no relational computation at all, which independent randomization rules out. At every depth including the final layer, all four targets are decodable at or below their own majority-class base rate (exact position match: 0.757–0.761 vs. a 0.788–0.790 base rate; loose match: 0.870–0.874 vs. 0.912–0.915; shape match: 0.817–0.843 vs. 0.833–0.837; color match: 0.828–0.839 vs. 0.830–0.837) – no lift from chance at any depth, for either rule. We do not read 0.870 against a 0.85 threshold as "encoded" here: that threshold implicitly assumes a roughly balanced target (as `touches(o)` is); against a target with a 0.91 base rate, 0.870 is worse than predicting the majority class and should not be credited as encoding.

A.11 A FIRST TEST OF MECHANISM-CONDITIONED RECOVERY: INTERVENTION COMMUTATIVITY UNDER SINGLE-HEAD ABLATION

Every result reported so far is behavioural recovery, $f(x) \rightarrow S_R$ (Section 3): a structure fit to a model's *output*, never checked against the model's mechanism M_x. Intervention commutativity requires translating a mechanism-level intervention into a structural edit, $\alpha_I : I_M \rightarrow I_S$, then checking $f(\mathrm{do}(M_x, i)) \approx h(\mathrm{do}(S_x, \alpha_I(i)))$; here α_I is instantiated as a small, pre-registered set of AST edits and tested against Oracle at 3 seeds on two rules (`every-circle-left-of-triangle`, L1; `worked-example-conditional`, L2; full edit sets and criteria in the appendix).

Of 6 (rule, seed) cells, 5 are vacuous under the mechanism-localization script's own causal-relevance criterion (agreement with baseline < 0.95 under single-head zero-ablation) – there is no mechanism-level intervention to test commutativity against. The one testable cell

(every-circle-left-of-triangle, seed 2) does not support the criterion: the best-matching pre-registered edit agrees with the ablated model's behaviour 79.4% of the time, *less* than the unedited structure's own 87.5% null-control agreement – no candidate structural edit tracks this ablation's effect better than making no edit at all. This first attempt at the full $S_x = \alpha(M_x)$ claim therefore finds no supporting evidence, narrowing what PSR's current mechanism-level tooling can establish for the rules in this library; whether a stronger intervention primitive would change this is not tested here.

A.12 Boundary-adversarial retraining: full numbers

Section 4.5 summarizes this experiment; Figure 9 gives the numbers behind it. at-least-one-touching-pair ($\exists x, y$: touches(x, y), L3) includes near-miss scenes directly in training, at comparable magnitude to true/false examples and matched object counts across splits (mean 11.7/11.5/9.7, ruling out an object-count shortcut). SlotAttn-Tr's standard-split accuracy (left panel) is not the clean confirmation a training-boundary-coverage explanation would predict: near-miss reaches ceiling, but plain false sits at exact chance and true is well below ceiling – consistent with a shortcut tied to the near-miss construction's specific geometric signature, not a general touching concept. A direct hand-built boundary-adversarial construction on the same rule ($n = 60$/category, 3 seeds; right panel) reaches mean ground_truth_fidelity $= 1.0$ for all 3 models, but decision_fidelity separates them sharply – only Oracle is consistent enough ($\approx$ 0.98–1.00) for the construction to be diagnostic rather than redundant with an already-noisy baseline, and even Oracle scores 0.067 ($n = 60$, 95% CI $[0.026, 0.159]$) on a tight-contact pair among non-touching distractors – as low as any rule whose training split saw no boundary-adjacent data at all.

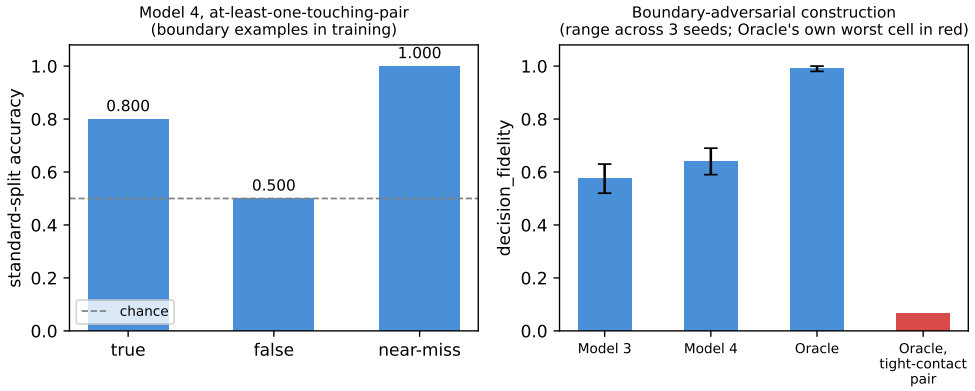


Figure 9: Left: SlotAttn-Tr's standard-split accuracy on at-least-one-touching-pair after boundary-adjacent training. Right: decision_fidelity range (3 seeds) on the direct boundary-adversarial construction, plus Oracle's own worst cell.

A.13 Hypothesis discrimination on the showcase rule: further nuances

Section 4.3 reports the headline attribution numbers (H_cluster at 97.0% for Oracle's shape-dependence-of-contact errors, region attribution for VQVAE/SlotAttn-Tr). Two additional scenes, built specifically so the region-membership alternatives predict differently from each other and from the true rule, sharpen the region-gating finding: VQVAE/SlotAttn-Tr's errors there match "ignore region entirely" and "use $x<1/2$" at 100% each in the respective scene built to isolate them, and an existing boundary category (left_region_boundary_just_outside) agrees at 100% with "ignore region entirely" for both models – their region-gating is measurably looser than the exact rule, without this test alone certifying *which* single looser boundary is responsible, since a model consistent with one such alternative in a given scene is, in that scene, indistinguishable from a model consistent with a family of similarly loose alternatives. A second nuance, not in the main text: the shape-blind contact alternative explains an identical 11/24 (46%) share of the tight-contact near-miss category's errors for *all three* models, Oracle included, despite Oracle having no continuous-distance

computation to blindly approximate with – read as this alternative's prediction coinciding with the true rule's own hardest boundary cases by construction, not as evidence of shape-blindness in Oracle specifically. The approximate-equality alternative explains *none* of any model's errors (0%, every category, all three models), ruling it out cleanly rather than leaving it merely untested; the decoy and off-by-one arithmetic categories remain unexplained by any of the five alternatives for any model.

A.14 Generalizing the boundary-adversarial pattern across the rule library

Section A.8's boundary-adversarial construction rested on a single rule. The same methodology – hand-built scenes placed directly at a rule's own decision boundary, each verified against `rule.evaluate()` before scoring, none used for training or selection – was applied to 7 further rules spanning L1–L4, at $n = 60$ per category (3 seeds $\times$ 20 instances), using existing multi-seed checkpoints for 5 of them and newly trained checkpoints (bringing `at-least-one-touching-pair` and `mirror-symmetry` up to the same 3-seed $\times$ 3-model coverage as the rest of the library) for the remaining 2.

Rule	Boundary category	VQVAE	SlotAttn-Tr	Oracle	SlotAttn-RN
`radial-symmetry`	tolerance, just inside	1.000	1.000	1.000	1.000
	tolerance, just outside	**0.000**	**0.000**	**0.000**	**0.000**
`mirror-symmetry`	tolerance, just inside	0.950	0.983	0.950	0.967
	tolerance, just outside	**0.033**	**0.050**	**0.067**	**0.033**

`mirror-symmetry` (mean `ground_truth_ fidelity=` 1.0 across all 3 models too) replicates `radial-symmetry`'s tolerance-boundary collapse closely (both drop from $\approx$0.95–1.00 to $\approx$0.03–0.07 across all 3 models, each outside the $[0.000, 0.060]$ 95% CI the `just-inside` side's near-ceiling accuracy would predict under no boundary effect at all), confirming this is a property of exact-tolerance Gestalt predicates in this library generally, not an artifact of `point_symmetric` specifically; both rules additionally show near-zero accuracy (0.000–0.37) when a candidate pair sits at the exact mirror point but violates the rule's `same_shape`/`same_color` conjuncts, indicating models rely on an approximate positional layout rather than checking either the exact tolerance or the type-matching condition the rule actually requires.

The same construction on L1 (`left-red-implies-right-blue`) and both L2 arithmetic variants (`red-count-triples-yellow-triangles`, linear; `yellow-count-mod-four-equals-blue-count`, periodic) finds real but less uniform collapses – each model has its own idiosyncratic boundary blind spot rather than all 3 failing identically, and for the periodic rule, Oracle's failure is concentrated specifically at higher-count wrap points (yellow$=4, 5$) while it remains exact at the lowest (yellow$=0, 1$), consistent with a memorized small-count lookup table rather than genuine modular arithmetic. `parity-five-indicators` (L4), tested via single-indicator flips rather than a continuous boundary construction (parity has no graded margin the way a threshold- or tolerance-based rule does: flipping any one indicator changes the answer, so there is no near-miss to build), shows no collapse pattern in the `ground_truth_fidelity` sense above, consistent with these models never reaching near-perfect standard-split fit on L4 rules in the first place – there is no "perfect fit" for a minimal-pair sensitivity test to collapse from. Applying the edit taxonomy of Appendix A.6 anyway yields two candidates: `H_xor4` (drop-a-conjunct: parity of only the first four indicators, ignoring `(square,red)`) and `H_majority` (change-the-aggregation-function: at least 3 of the 5 indicators present, rather than an odd count). Both are scored the same way as every other rule's AM (Section 3) against the same frozen checkpoints, on four categories of $n = 300$ single-indicator-flip trials each (`flip_add_baseline`, `flip_add_flipped`, `flip_remove_baseline`, `flip_remove_flipped`; torch seed 1234 for SlotAttn-Tr). Pooled across all four categories, neither alternative clears the 0.80 identifiability bar (`H_majority`: 0.485/0.542/0.493 for VQVAE/SlotAttn-Tr/Oracle; `H_xor4`: 0.453/0.534/0.502) and the label recorded in `profiles.csv` (Appendix A.6) stays *unexplained*, unchanged. The pooled number is diluted the same way pooling dilutes AM elsewhere in this library (Section 4.3): on the two categories where majority-vote and true parity agree by construction (`flip_add_baseline`, `flip_remove_baseline`), a wrong prediction cannot match `H_majority` and contributes nothing but dilution. Restricted to the two categories where majority-vote and true parity structurally diverge (`flip_add_flipped`, `flip_remove_flipped`) – i.e. where every scene is constructed

so H_majority and the true rule predict opposite labels – H_majority matches *every single* wrong prediction, for all three architectures (303/303 VQVAE, 321/321 SlotAttn-Tr, 298/298 Oracle). We report this number for completeness but do not read it as evidence: for a binary label, once a category is built so the two hypotheses always disagree, any wrong prediction is *mathematically forced* to match whichever hypothesis is not the true rule, independent of the model's mechanism – 303/303 is guaranteed by the construction, not an empirical finding about any of the three architectures. (This is the general version of the AM-on-errors caveat in Appendix A.18: scoring only wrong predictions against a hypothesis that disagrees with the truth on all or most of the scenes being scored can manufacture a high match rate with no mechanistic content; here the effect is total rather than partial, because the category is built for exactly that disagreement.) The pooled label – computed on the full four-category set, where the two hypotheses do *not* always disagree – remains *unexplained* for all three architectures, and that is the number this paper's criterion counts.

A.15 Replication on Geometry-Corrected Data

As a robustness check on the boundary-adversarial results, we regenerated the selected replication tasks using exact per-shape canvas geometry in the scene sampler (each shape's true rendered circumradius, rather than a fixed fraction of its size) and, for left-red-implies-right-blue specifically, a full-containment region predicate (left_region_full/right_region_ full: an object's entire rendered body inside the region, not only its centroid) in place of the centroid-only predicate used elsewhere in the library. The eight configuration-based rules included in this replication – the 8 core testbed rules (Section 2) with touching-count-matches-red-left excluded and majority-red-of-four included instead – had their train/val/test splits regenerated at the same per-split sizes as the existing configs (configs/rules/*.yaml); red-triangle-proportion-ge-20pct was regenerated the same way but through its own dedicated generator (gen_proportion_threshold.py) rather than a configs/rules_geomfix/ YAML file, consistent with how it is trained outside the geometry-corrected check too (Appendix A.20). Every scene, across both paths, was verified to have zero canvas-edge violations, zero real object-object overlaps, and a stored label matching a fresh rule.evaluate() call, across all 920–1836 scenes per rule. As a check separate from each architecture's standard per-rule training regime (Appendix A.21), all four architectures (VQVAE, SlotAttn-Tr, Oracle, SlotAttn-RN) were retrained from scratch at 3 seeds each on this newly regenerated data (scripts/run_geomfix_full_retrain.sh), using the same training scripts and default hyperparameters as the original checkpoints – seed is the only varied factor, exactly as in Appendix A.14.

The central finding replicates. The same boundary-adversarial stress-test scripts (same instance-seed convention, n=60 per category) were run against the newly trained checkpoints for 8 of the 9 rules (red-count-triples-yellow-triangles, yellow-count-mod-four-equals-blue-count, parity-five-indicators, radial-symmetry, mirror-symmetry, at-least-one-touching-pair, majority-red-of-four, red-triangle-proportion-ge-20pct (VQVAE/SlotAttn-Tr/Oracle only, consistent with this rule's published CF/AM numbers not using a SlotAttn-RN checkpoint either, Appendix A.6); left-red-implies-right-blue is treated separately below, since its region predicate itself changed. Of 126 directly comparable (rule, boundary category, model) cells, 17 stay collapsed (< 0.20 accuracy) on both the original and the corrected data, 32 stay near-ceiling (≥ 0.80) on both, and 57 stay in between on both – 106 of 126 cells not just unchanged in direction but in the same bucket. The remaining 20 shift between adjacent buckets (collapsed↔ in-between or in-between↔near-ceiling) under ordinary retraining variance (individual cells move up to ± 0.45 at n=60 per cell and fresh random initialization, roughly as many up as down); *zero* of the 126, including those 20, cross the full gap between collapsed and near-ceiling in either direction, so no rule's qualitative conclusion changes.

left-red-implies-right-blue's own boundary-adversarial categories are not directly comparable to the original numbers, since its region predicates changed from centroid to full-containment (a different decision boundary, not merely cleaner data for the same one); we report the new numbers on their own instead. On the corrected data and semantics, Oracle's boundary results illustrate the collapse pattern directly: 0.967 accuracy on antecedent_boundary_just_inside (antecedent correctly fires) collapses to 0.000 on antecedent_boundary_just_outside (n=60 each) –

Oracle answers "antecedent fires" for every scene tested near this boundary regardless of which side of the full-containment line the object actually falls on, rather than tracking the boundary itself. VQVAE and SlotAttn-Tr show the same qualitative asymmetry at smaller magnitude ($0.500 \rightarrow 0.467$ and $0.800 \rightarrow 0.117$ respectively).

An unrelated finding on the Ki-67-motivated rule. This subsection uses a rescaled construction of `red-triangle-proportion-ge-20pct` – 120 triangles total (count(red triangle) ≥ 24) rather than the 60-triangle, 12-triangle-threshold version Section 4.1 and Appendix A.6 report `CF`/`AM` numbers for – built purely to get a finer-grained sweep of exact counts around the same 20% cutoff; it is not the dataset behind this paper's published `CF`/`AM` numbers for this rule. Scanning across a range of exact red-triangle counts (14–34, 3 fresh scenes per count, all four architectures at 3 seeds each) lets us reconstruct each trained model's *implicit* decision threshold directly, rather than only testing accuracy at the true boundary ± 1. SlotAttn-Tr and SlotAttn-RN each converge to the same implicit threshold across all 3 of their own seeds (25, a consistent off-by-one shift). VQVAE's three seeds land at 22, 22, and 20. Oracle – exact symbolic input, no perception in the loop at all – is the *least* consistent of the four: its three seeds' implicit thresholds are 26, 23, and 20, a 6-count spread that brackets the true value of 24 without any seed landing on it. These are aggregate tendencies over independently drawn scenes per count, not a claim about any one fixed image: individual scenes at the same count already disagree substantially seed-to-seed in the underlying scan (full curve in the released logs). Figure A.15 makes the resulting cross-architecture disagreement concrete on one specific scene, separately selected (from a pool of scenes at this exact count that different architectures do *not* all agree on, not a random or representative draw) to show that fixing the image entirely does not fix the disagreement: at the exact clinical cutoff ($24/120 = 20.0\%$, ground truth: high risk), SlotAttn-Tr and SlotAttn-RN unanimously classify this particular scene correctly (3/3 seeds) while VQVAE's majority vote (2/3 seeds) clears the same image as low-risk – one concrete instance of the disagreement the aggregate thresholds above describe in general, not a typical or average case for any one architecture. This is not a property of the rule being unrecoverable from data in principle: the same enumerate-and-score program synthesis used throughout this paper (Section 3), fit directly against `rule.evaluate()` on all 920 scenes with no model in the loop and no privileged knowledge of the generating rule, recovers `Count(red ∧ triangle) ≥ 24` at accuracy $= 1.0000$ – the exact threshold, unambiguously the top-scoring candidate (next best: 0.9913, a structurally unrelated count-compare template; the same template at thresholds 25 and 26 score 0.9902 and 0.9859, correctly ranked below the exact match). Program synthesis over ground-truth labels locates the correct threshold exactly; no individually trained architecture's implicit threshold, at any seed, does.

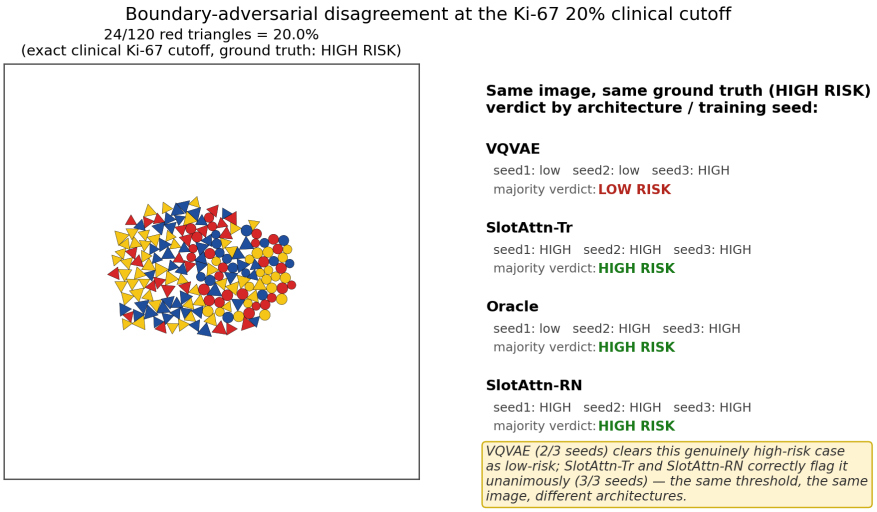


Figure A.15: one scene at the exact Ki-67 20% cutoff (24 of 120 red triangles), ground truth high-risk, selected from a pool of same-count scenes on which the four architectures disagree (not a typical or average scene for any one of them – see the aggregate implicit-threshold numbers in the main text for that). On this particular image, VQVAE's majority vote across seeds misclassifies it as low-risk; SlotAttn-Tr and SlotAttn-RN classify it correctly, unanimously across all 3 seeds.

A.16 A FIFTH, ARCHITECTURALLY DISTINCT MODEL: PIXEL-SPACE DIFFUSION

Every model above shares a common ancestor: a relational Transformer Stage 2 (VQVAE, SlotAttn-Tr, Oracle) or a Relation Network built to mirror it on the same frozen Slot Attention front end (SlotAttn-RN). As a check that the boundary-collapse pattern is not an artifact of that shared lineage, a fifth model – a conditional pixel-space DDPM trained directly on rendered images, with no object-centric or tokenized perception stage of any kind – is added, classifying via the standard noise-prediction-error trick: for label $y \in \{\text{True}, \text{False}\}$, predict $\arg\min_y \mathbb{E}_{t,\epsilon}\|\epsilon_\theta(x_t, t, y) - \epsilon\|^2$, estimated by Monte Carlo over 16 paired (t, ϵ) draws per scene (both labels scored on the same draws to reduce variance). Trained from scratch per rule at 3 seeds, all 9 rules, on the same training images used for VQVAE, SlotAttn-Tr, and Oracle on each rule (`touching-count-matches-red-left` itself was not included in this model's training sweep and has no checkpoint; for `red-triangle-proportion-ge-20pct` specifically, that means the same generator-produced 60-triangle/12-threshold images used to train those three models, Section 4.1 – the construction this paper's published `CF/AM` numbers for this rule come from, and the one with no SlotAttn-RN checkpoint; the separate 120-triangle/24-threshold construction in Appendix A.15 above *does* have SlotAttn-RN checkpoints, trained only for that implicit-threshold check).

Validity-gate accuracy is weak and uneven, unlike the rest of the roster. Standard-split classification accuracy, 3 seeds each:

Rule	Accuracy (mean of 3 seeds)
mirror-symmetry	0.972
red-triangle-proportion-ge-20pct	0.913
radial-symmetry	0.869
majority-red-of-four	0.854
left-red-implies-right-blue	0.610
red-count-triples-yellow-triangles	0.546
at-least-one-touching-pair	0.537
yellow-count-mod-four-equals-blue-count	0.498
parity-five-indicators	0.479

Mean 0.698 across all 9 rules – far below VQVAE/SlotAttn-Tr/Oracle's near-1.0 standard-split fit, and below our own 0.95 validity-gate bar on 8 of 9 rules (only mirror-symmetry clears it). Under the training budget used here (matched to the other four models, not separately tuned for this architecture), a pixel-space diffusion classifier does not reliably learn most of these rules at all – a limitation of this check, not only a boundary-specific finding: for the 5 rules below 0.85, a boundary-adversarial test is not diagnostic, since the model has not first established a clean fit for the boundary to be adversarial *to*.

Boundary-adversarial results on the best-fit rule. We ran the same boundary-adversarial stress-test suite used for the rest of the roster on the one rule with the highest validity-gate accuracy, where a boundary result is actually interpretable. `mirror-symmetry` (0.972 validity gate, the one rule clearing our bar) shows a clean replication of the same collapse pattern:

Category ($n = 60$ each)	Accuracy
shape/color decoy, color mismatch	0.717
near-symmetric decoy	0.633
tolerance boundary, just inside	0.883
tolerance boundary, just outside	0.133
shape/color decoy, shape mismatch	0.050

The shape-mismatch category – the direct analogue of shape-dependence-of-contact, the constituent every other architecture also fails on this rule (0.04–0.33 pooled CF, Appendix A.6) – collapses to 0.050 here, if anything more severe than the rest of the roster, on an architecture sharing no perception or relational-Stage-2 lineage with any of them. The tolerance-boundary asymmetry (0.883 inside vs. 0.133 outside) is the same qualitative signature (near-ceiling on one side of a threshold, collapse on

the other) the equality/threshold constituent shows for the other four models, though we report the two sides separately rather than pooling them, for the same reason argued in Section 4.2.

Reading of the confound-isolation question. On the one rule where this model actually learns the task well enough for a boundary test to mean anything, the collapse pattern replicates, on an architecture with no perception or relational-Stage-2 lineage shared with any other model in this paper's roster – evidence against the collapse being an artifact of that shared lineage, on this rule. We do not claim this generalizes across all 9 rules: for 8 of them, the validity gate itself fails, and a boundary-adversarial result on a model that has not learned the rule at all is not informative about boundary-specific failure, only about the model's overall unreliability under this training budget. A properly powered version of this check (longer training, more capacity, or both, tuned per rule until diffusion classifiers reliably clear the validity gate the other four architectures already clear) is future work, not attempted here under this paper's timeline.

A.17 Full Alternative-Hypothesis Catalogue

Figure 3 (Appendix A.6) plots the full explanation profile referenced from Section 4.3. Five alternatives are scored against every model's errors on `touching-count-matches-red-left`: `H_dist`, a shape-blind distance in place of true contact; `H_allred`, which drops the region conjunct entirely; `H_cutoff`, which moves the region boundary from $x<1/3$ to $x<1/2$; `H_approx`, which tolerates an off-by-one count mismatch; and `H_cluster`, which counts touching *clusters* rather than touching objects. The oracle's shape-dependence failures attribute to `H_cluster` at 97.0% ($\texttt{CF} = 0.196$), crossing our bar for *behaviourally consistent with* this alternative – the one case in this catalogue additionally validated on held-out instances (Section 4.3). The same alternative shows 100% error attribution on the region constituent for both image-based models, but this is raw error overlap, not holdout-tested, and the holdout test that does exist for this rule finds `H_cluster` a significantly *worse* account of VQVAE's and SlotAttn-Tr's decisions than the true rule (Appendix A.18); it does not explain the oracle's own region failures either: there, $\texttt{CF} = 0.750$ is too high to cross that bar even though `H_cluster`'s attribution is again 100%. Which constituent breaks, in other words, depends on the model, not only which alternative accounts for it once it does. Aggregation resists explanation altogether – every alternative for every model leaves it *unexplained*, because the decoy categories were built so that `H_cluster` predicts the *same* label as the true rule there and so cannot discriminate. On `touching-count-matches-red-left` itself, none of the five alternatives explains the comparison constituent's boundary category at all ($0.000/0.000/0.083$).

The same scoring, applied to seven further rules, replicates this pattern rather than being specific to one (full numbers in Appendix A.6). On `radial-symmetry`, `H_ignore_type` and `H_loose_tol` each explain 100% of every architecture's errors, across the board. Its structural twin, `mirror-symmetry`, replicates the shape-dependence result but not the comparison one, where the same alternative's attribution dilutes to $\approx 48\%$ once pooled and stays *unexplained* – a reminder that an alternative generalizing within one rule does not guarantee it generalizes to a second rule the theory predicts should behave the same way. On `red-count-triples-yellow-triangles`, dropping the triangle-shape conjunct explains 100% of VQVAE's and SlotAttn-Tr's errors but not the oracle's. Three further rules show the mirror image of that pattern: 100% attribution on an individual category, diluting to *unexplained* once every category for that constituent is pooled together. Even `parity-five-indicators`, an SQ-hard k-way parity with no continuous boundary to speak of, yields two candidates from the same fixed taxonomy of edits; neither clears the pooled bar, but one matches every wrong prediction on the two categories actually capable of distinguishing it from the true rule (Appendix A.14). All eight rules carry `AM` evidence, not only the flagship one.

A.18 From Diagnosis to Prediction: Do Alternative Hypotheses Predict New Cases?

Every `CF`/`AM` number reported so far is computed on one pooled set of boundary-adversarial instances: the same instances used to determine which named alternative best explains a model's errors are also the instances that number is reported on. This leaves open whether an identified alternative is a predictive account of the model's behaviour, or a description that happens to fit the specific sample it was read off. We test this directly.

Calibrate on half the instances, predict the other half. For `touching-count-matches-red-left`, 6,795 scored boundary-adversarial instances (40 per category per seed, up from this rule's usual 8, specifically to support a clean split; 3 seeds, all three models) are split by an interleaved instance index into a calibration half and a disjoint holdout half never touched during calibration. On the calibration half, the best-matching alternative hypothesis is identified exactly as elsewhere in this paper (highest fraction of wrong predictions matched); that same, already-fixed hypothesis is then scored – with no further fitting – on the holdout half.

Model	Calibration match	Holdout match	Gap	H_{cluster}'s own error rate	Holdout p
VQVAE	0.524 (n=418 wrong)	0.542 (n=417)	+0.018	0.738	1.9×10^{-29}
SlotAttn-Tr	0.514 (n=484)	0.522 (n=456)	+0.007	0.738	1.1×10^{-42}
Oracle	0.809 (n=614)	0.820 (n=615)	+0.010	0.738	9.1×10^{-12}

A flat 0.5 chance floor is the wrong null here, and using it inverts part of the conclusion. `AM` is computed only on scenes where a model is wrong; for a binary label, a wrong prediction is definitionally the *other* label, so on that subset $\texttt{AM}(c, m, a) = \Pr[a(x) \neq y(x) \mid f(x) \neq y(x)]$ – *by construction*, a property of how often alternative a itself disagrees with the true rule on the scenes being scored, not directly a mechanistic claim about the model. This matters concretely here because `H_cluster` is not a coin flip relative to the true rule on this pool: it disagrees with it on 73.8% of all $6{,}795$ boundary-adversarial scenes (a property of the hypothesis and the scene pool alone, independent of any model), so *any* model whose errors were uniformly distributed across this pool, for reasons having nothing to do with cluster-counting, would already show $\texttt{AM}(\texttt{H_cluster}) \approx 0.74$, not 0.5. Testing the observed holdout match against a flat 0.5 null therefore overstates how informative the oracle's number is and, worse, mischaracterizes VQVAE's and SlotAttn-Tr's numbers as an uninformative coin flip when they are not: scored with a hypergeometric test against `H_cluster`'s own 73.8% base rate as the null (are matches concentrated above or below what an error set unrelated to `H_cluster` would already produce, given its size and this base rate), all three models are highly significant, in *different directions*. All three independently select `H_cluster` (aggregation: touching *clusters* rather than touching *objects* – unrelated to this rule's region predicate, so not confounded by the centroid-vs-full-containment issue Appendix A.15 documents for a different rule) as the calibration-half winner, and the holdout match rate does not degrade for any of them (replicated separately in the calibration half too, $p = 2.1 \times 10^{-34}$/1.7×10^{-50}/1.4×10^{-9} for VQVAE/SlotAttn-Tr/Oracle). But only the oracle's match sits *above* the 73.8% base rate (82.0%, a real but modest ≈ 8-point lift, not the 82%-vs-50% contrast a flat null would suggest); VQVAE's and SlotAttn-Tr's holdout matches (54.2%, 52.1%) sit significantly *below* it – their errors are explained by `H_cluster` *less* often than an unrelated error set of the same size would be by chance, a replicated finding in its own right, not the "open question" a $p = 0.5$ null made it look like.

A second, independent measure sharpens the same split. `decision_fidelity` scored against *every* decision on the same holdout half, not only the model's errors, is the most direct test of whether a hypothesis is a good overall account of a model's behaviour: Table 2 in Section 4.3 reports it.

For the oracle, `H_cluster` is not merely the best-attributed alternative on errors – it is the single best account of *all* of the oracle's decisions among the six hypotheses tested, exceeding even the true rule's own decision fidelity (0.610 vs. 0.457). For VQVAE and SlotAttn-Tr, `H_cluster` is the *worst* of the six by the same measure, well below their own true-rule decision fidelity (0.293/0.279 vs. 0.632/0.595): a general account of their decisions worse than chance-level guessing would give, not merely a weak one. Read together with the corrected null above, the two measures agree in direction: this is a statistically significant predictive result for the perception-free oracle specifically (a real, if modest, positive association, replicated calibration-to-holdout, that also improves its overall decision account); for VQVAE and SlotAttn-Tr, both measures agree in the *opposite* direction – `H_cluster` is a significantly worse account of their errors and their overall decisions than an unrelated baseline would be, a confirmed negative finding, not an open question. This is the direct test of whether a recovered explanation *predicts* a model's behaviour on new, deliberately distinguishing cases, rather than only redescribing the cases it was read off – for the oracle, it does, in a smaller way than a naive chance comparison suggested; for the other two, it does not.

Figure 1 in the Introduction makes this concrete on a single held-out scene: 2 separate touching pairs give 4 objects that touch something but only 2 connected clusters, so the true rule (object count, `TRUE`) and `H_cluster` (cluster count, `FALSE`) disagree, and the oracle sides with `H_cluster`.

Is the same constituent linearly present in the two image-based models? Section 4.4's oracle probes rely on a fixed object-to-token correspondence its patch/slot-based counterparts do not have, so only the per-scene targets (needing no such correspondence) transfer: mean-pooled representations at all three depths, 3 seeds each, same held-out boundary pool as the calibration/ holdout test above. Rounded-to-nearest-integer accuracy against `n_touching_clusters` exceeds accuracy against `n_touching_objects` at every depth, every seed, for both models (3-seed-mean accuracy, VQVAE: 0.22–0.31 vs. 0.44–0.56; SlotAttn-Tr: 0.21–0.26 vs. 0.44–0.50) – the same direction as the oracle's own asymmetry. The resemblance is only partial, though: the oracle's 83.9% `n_touching_clusters` accuracy clears the scene pool's 56.1% majority-class floor (the fraction of scenes with exactly one touching pair), while neither image-based model's does – their 3-seed-mean cluster accuracy is at or below this floor at every depth, and object accuracy is below it by a wider margin. (Raw mean-absolute-error favors `n_touching_clusters` even more starkly, but this is mostly an artifact of its roughly half-as-wide value range and mostly disappears once normalized against a constant-predictor baseline – see `DISCREPANCIES.md` §5.1 for the full check.) We read this as a small, consistently-replicated relative preference for cluster-counting over object-counting in these two representations, but not as evidence that either concept is confidently linearly encoded above trivial guessing – a more equivocal picture than the oracle's, closer to "not robustly encoded" than to "linearly decodable despite task failure" for this constituent specifically.

A.19 Does a feature-attribution baseline see this failure?

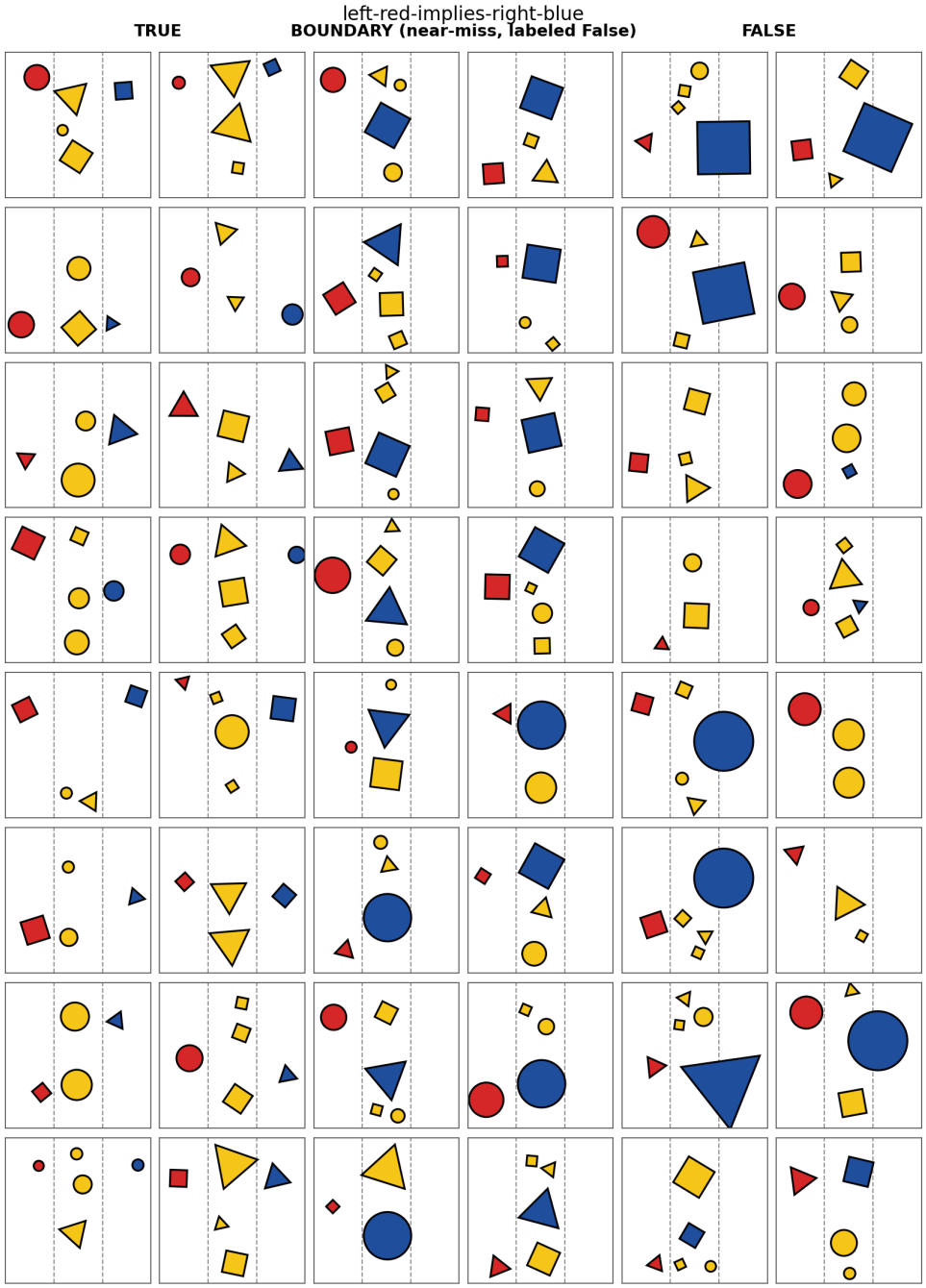
Section 5 argues in principle that attribution methods have no coordinate for a relational constituent; here we make that concrete against the exact boundary failure above. the feature-attribution baseline script implements Gradient×Input (standing in for LIME/SHAP/Grad-CAM, which share its per-feature output space and its inability to name a relation; Concept Relevance Propagation [40] targets a different explanandum and is not run against this specific Oracle scene pair – the separate comparison we do run it against, on SlotAttn-RN, follows below) and is run against the same `touch_boundary_true` / `touch_boundary_false_near_miss` scene pair Section A.8 scores Oracle on, using the same trained checkpoint. On one representative pair: Oracle predicts *True* for both scenes (correctly for the touching case, incorrectly for the narrowest-non-touching-gap case – the boundary failure itself, reproduced directly). Gradient×Input's relevance for the manipulated pair's two objects changes substantially between the two scenes (e.g. one object's `rotation` relevance shifts by 0.91, its `x` relevance by 0.71) – but this change tracks the two constructions' different object coordinates, not the touch relation: the attribution's output space is one relevance score per object per unary feature (shape, colour, size, position, rotation), with no coordinate that could represent "objects 2 and 3 touch." Even where attribution shifts most, it provides no signal that would let a practitioner distinguish "the model correctly used a relational touching predicate" from "the model's prediction happens to correlate with these coordinates" – the level Structural Explainability is built to recover and feature attribution is not. This is a single illustrative instance, not a statistically powered claim; the conceptual point – that the attribution family cannot represent a relation as a first-class output regardless of instance – holds independently of how any one example comes out.

Does CRP see this failure, where it is actually runnable? Concept Relevance Propagation [40] requires an LRP backward pass, which is an open problem through the self-attention every other model in this library's relational Stage 2 uses (Section 3). SlotAttn-RN (Slot Attention perception frontend, a Relation Network [2] instead of the shared attention Stage 2, so no self-attention anywhere in its relational computation) is the one architecture in this roster CRP is actually computable on, so we ran it there: an epsilon-rule LRP backward pass through SlotAttn-RN's two Relation-Network MLPs, redistributing each linear layer's bias share rather than discarding it (the standard alternative convention, since discarding leaves 70–95% of relevance unaccounted for on this network), verified by relevance conservation ($|\sum_i R_i - \text{logit}|$ relative error 5.7×10^{-4} and 6.5×10^{-4} on two independent checkpoints, orders of magnitude tighter than the 4.07 and 1.18 relative error the literal, bias-discarding rule gives on the same scenes). We then took the CRP step proper – clustering per-scene relevance patterns (k-means, $k \in \{3, 4, 6\}$, on the Relation Network's pooled relation-

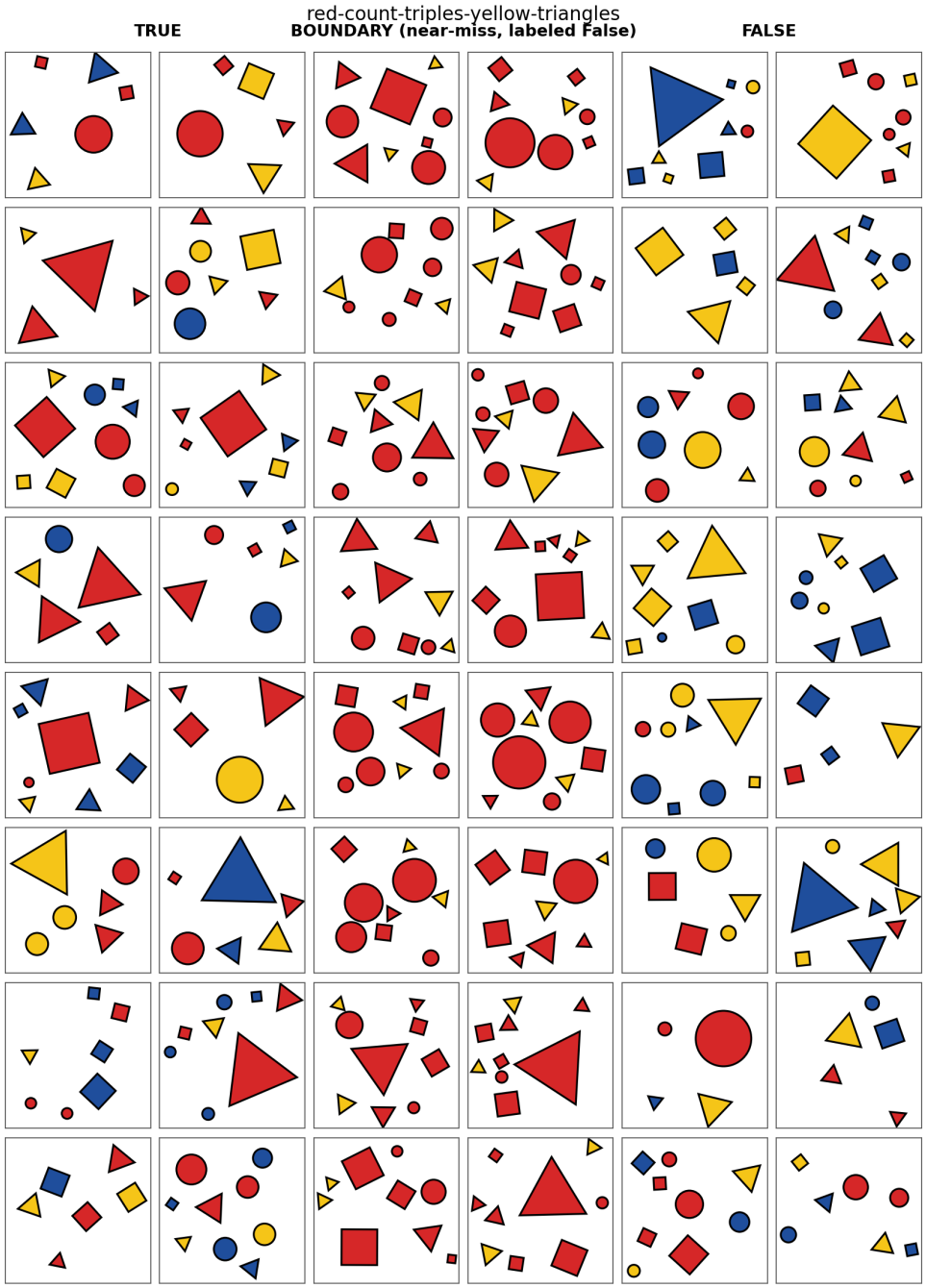
vector relevance) – on `at-least-one-touching-pair`, whose boundary categories are built as a matched True/False pair differing by the narrowest possible margin (Appendix A.8). The resulting concepts separate scenes by how much touching structure is present overall but do not separate the matched pair itself: the same cluster contains `touch_boundary_true` (29) and `touch_boundary_false_near_miss` (30) in near-identical numbers, consistent across every k tried. This is a real, checkable instance of the limitation Section 5 states in principle: a CRP concept, even computed correctly with verified relevance conservation, groups by how much relevant structure a scene contains, not by whether a specific named relational predicate (`touches` at its own tolerance boundary) is satisfied – the exact distinction `CF` is built to score, and the one CRP's clustering does not resolve here even where it is directly runnable.

A.20 Illustrative Examples for All Rules

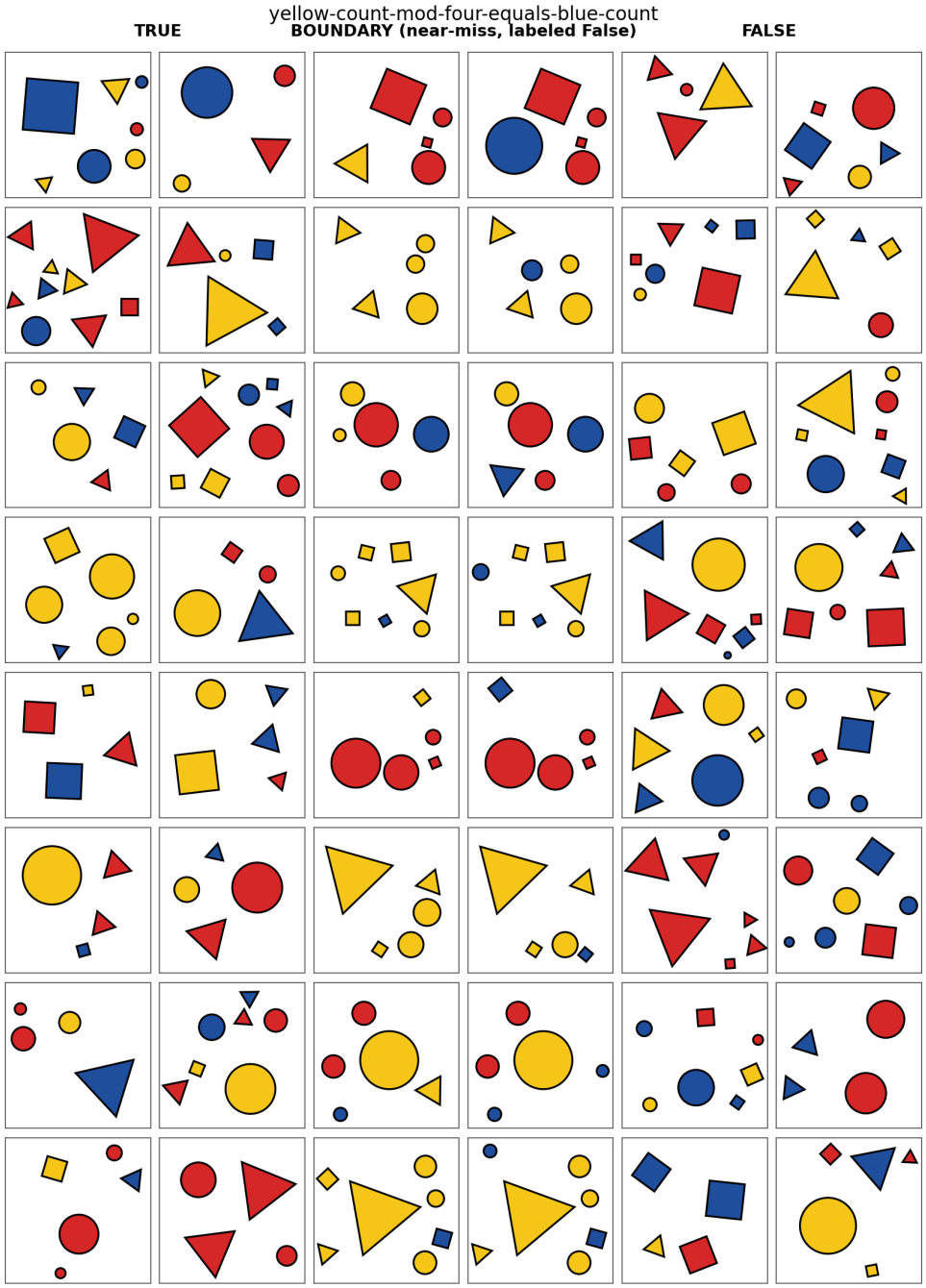
Appendix A.2 gives one True/False scene pair for `touching-count-matches-red-left`; for completeness, the same treatment for the rest of the rule library, rendered with the same object geometry used to train every model (`kandinsky_core/render.py`). Unless noted otherwise, every grid below follows the same layout: left block, 16 standard-distribution TRUE scenes; middle block, 16 instances of the rule's own tightest boundary-adversarial FALSE construction (Appendix A.8), a near-miss of the TRUE condition; right block, 16 ordinary standard-distribution FALSE scenes. Every one of the 48 scenes in every grid is verified against the rule's own `evaluate()` on the exact scene shown, not assumed from a directory or category name.



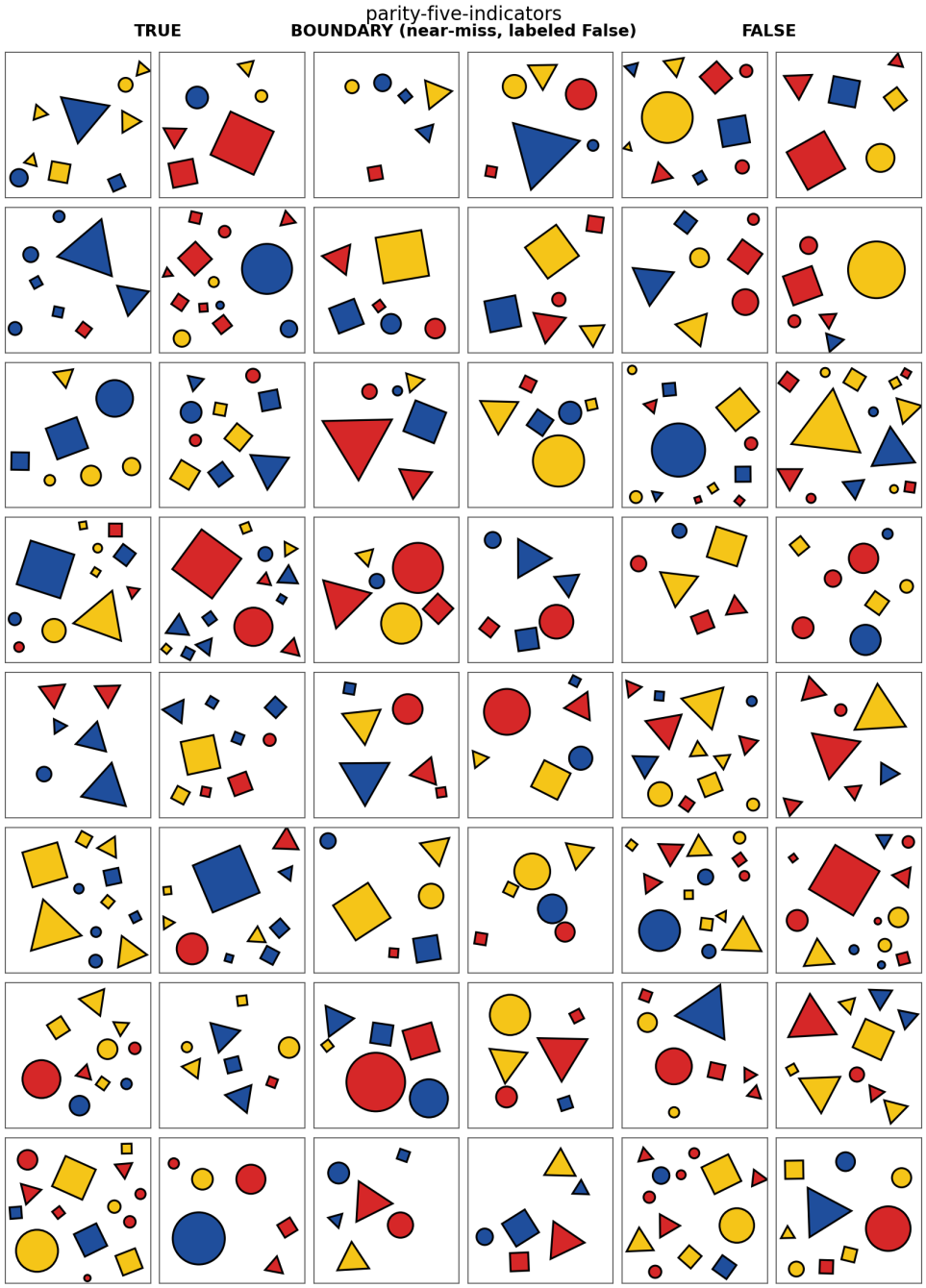
`left-red-implies-right-blue`: if a red object occurs fully within the left third, a blue object must occur fully within the right third (full containment, not centroid position; Appendix A.15). Middle block's near-miss: an object's edge sits a hair on the wrong side of the region line.



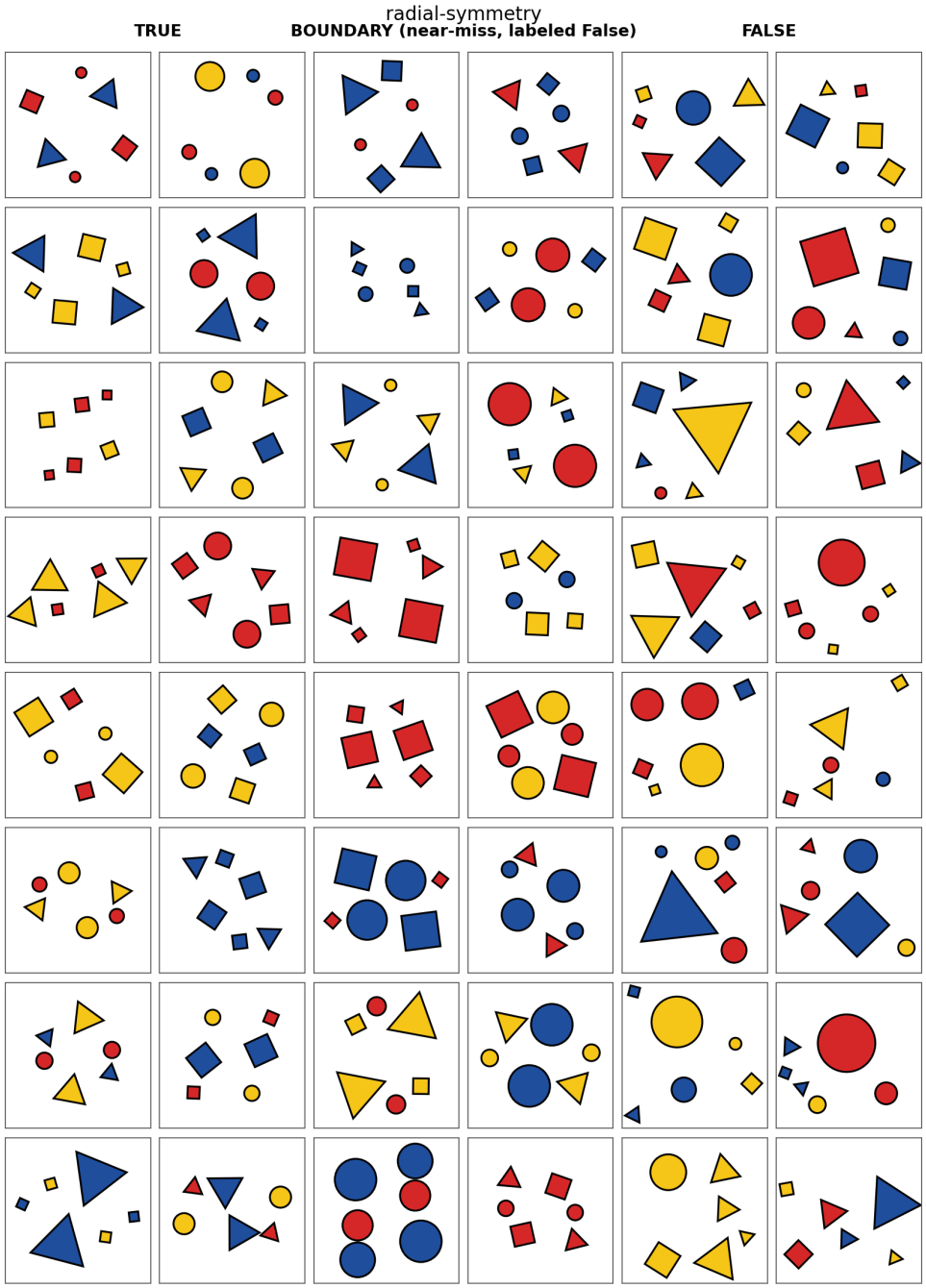
`red-count-triples-yellow-triangles`: count(red)= 3×count(yellow triangle). Middle block's near-miss: the ratio off by exactly one red object from the true 3× boundary.



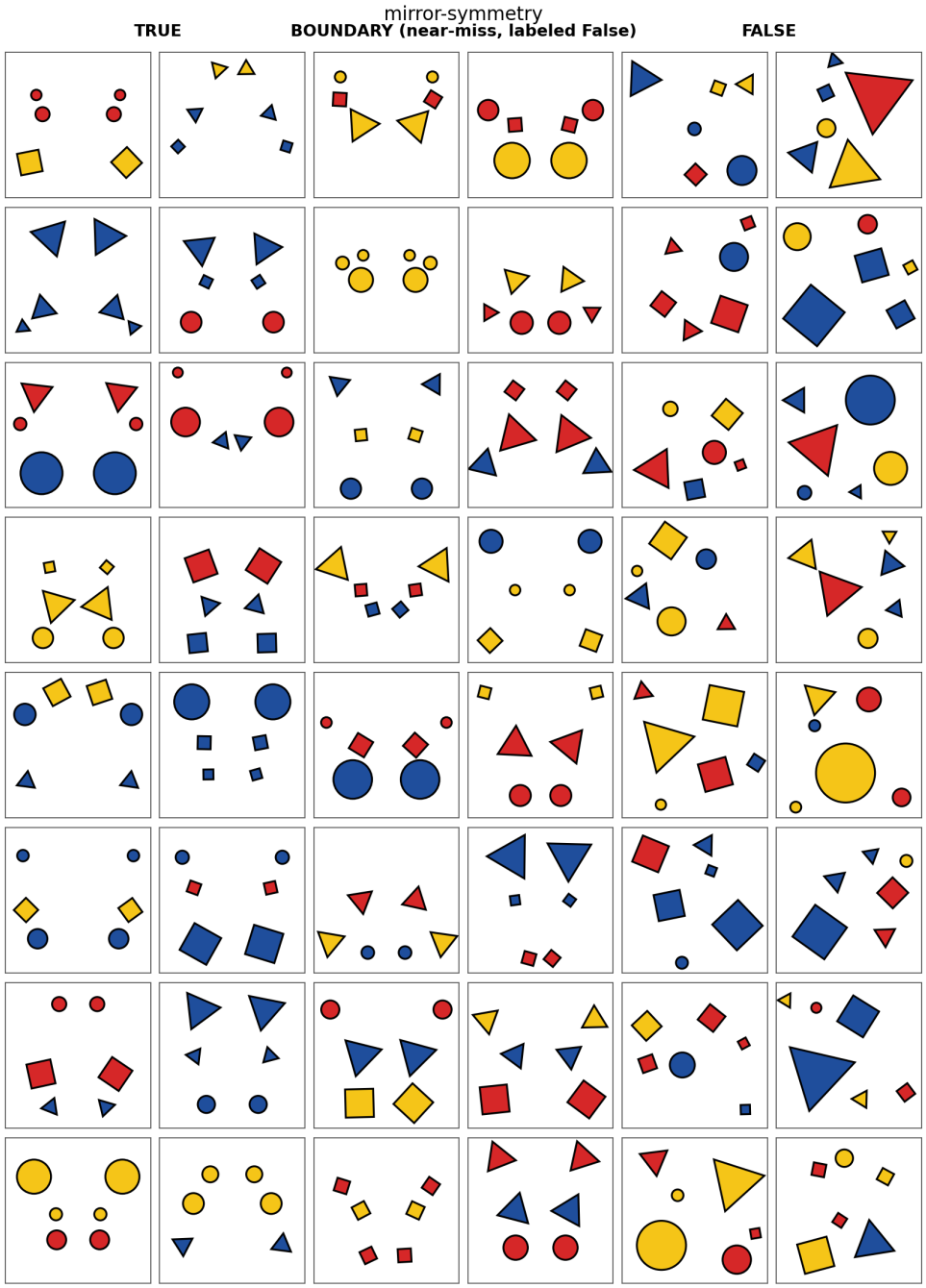
`yellow-count-mod-four-equals-blue-count`: count(yellow) mod 4 = count(blue), a genuinely periodic boundary that recurs every 4 objects rather than a single threshold. Middle block's near-miss: the mod-4 remainder off by one from the count of blue objects.



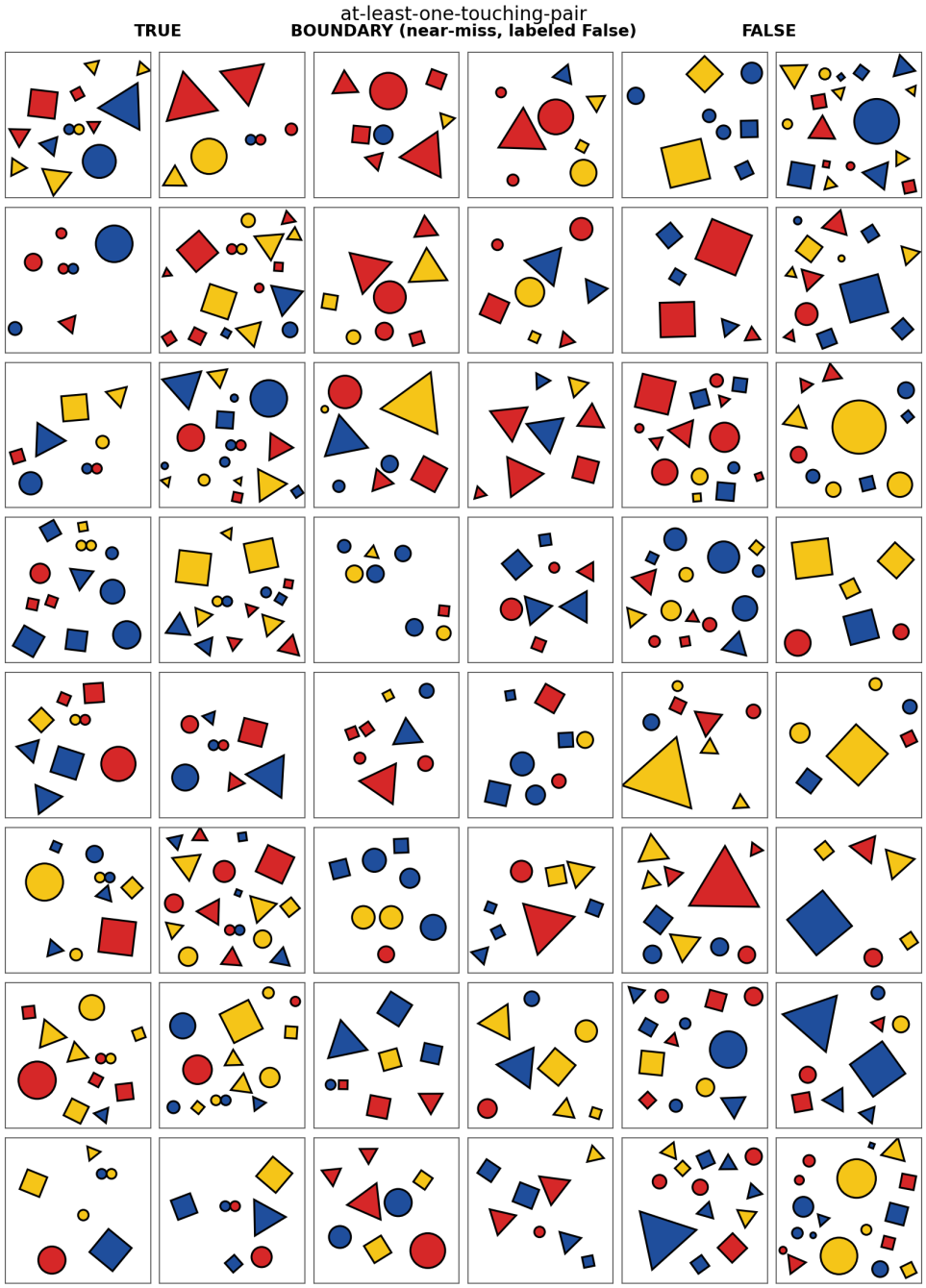
`parity-five-indicators`: an odd number of the 5 (shape,color) indicators {circle-red, square-blue, triangle-yellow, circle-blue, square-red} occur at least once. This rule has no continuous decision boundary at all (Section A.5.1), so the middle block's near-miss is instead a single-indicator flip: one of the 5 indicators toggled from present to absent or vice versa, which always flips the label.



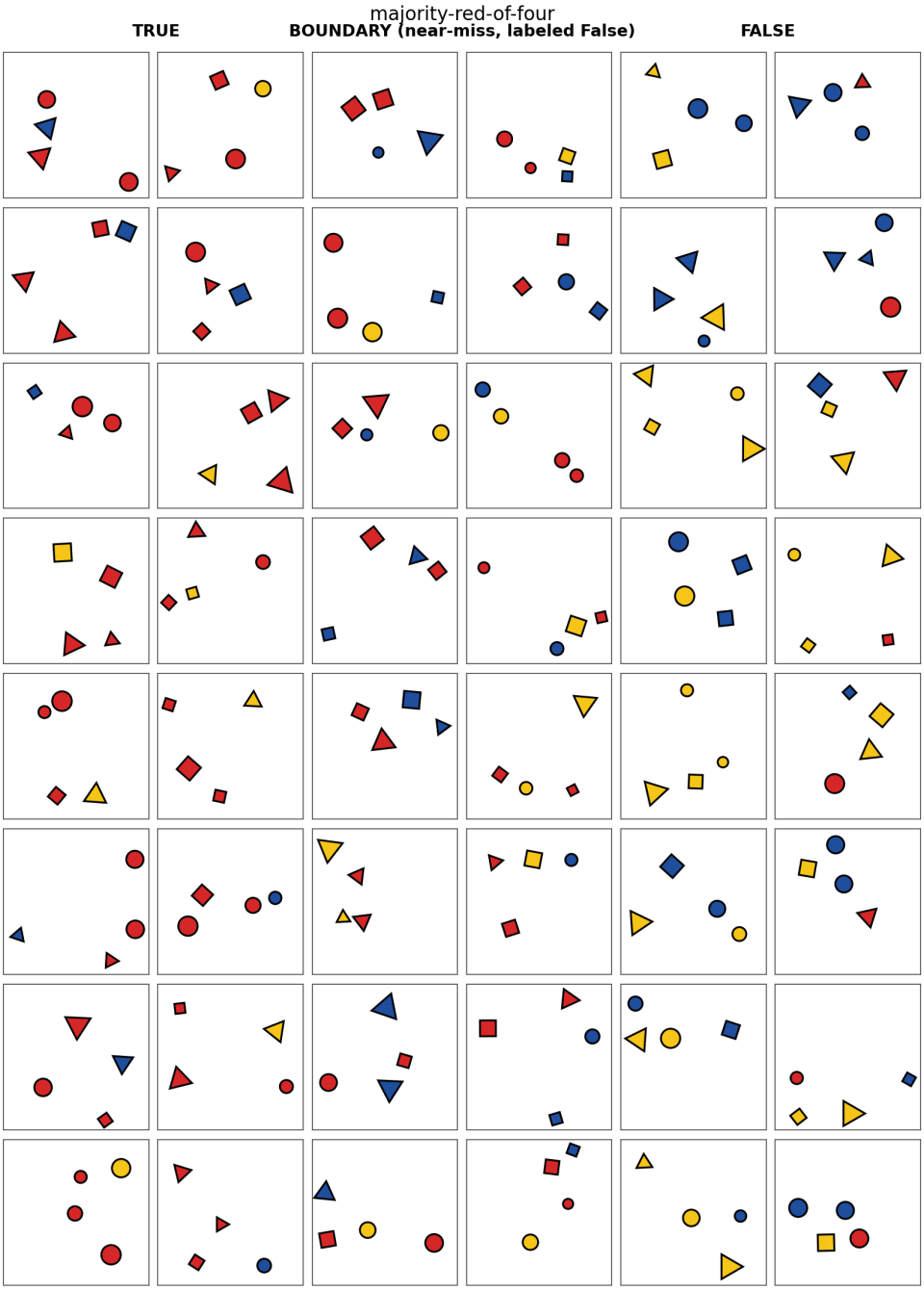
`radial-symmetry`: every object has a same-shape/same-color partner at the 180°-rotated position about the canvas center, within `point_symmetric`'s 0.02 tolerance. Middle block's near-miss: a partner placed just outside that tolerance.



`mirror-symmetry`: every object has a same-shape/same-color partner reflected across the fixed vertical axis $x = 0.5$, the same tolerance-based predicate as `radial-symmetry` but cheaper (a single coordinate, not a rotation). Middle block's near-miss: a partner placed just outside the mirror tolerance.



`at-least-one-touching-pair`: at least one pair of objects physically touches (`physics.touches`, tolerance 0.01) – the simplest `touches`-based rule in the library, no count or region confound. Middle block's near-miss: the narrowest possible non-touching gap.



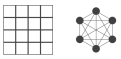
`majority-red-of-four`: at least 3 of exactly 4 objects are red – the positive-control rule added to check whether the audit can certify success, not only detect failure (Section 4.1). Middle block's near-miss: exactly 2 of 4 red, one fewer than the majority boundary.



`red-triangle-proportion-ge-20pct`: at least 12 of the 60 triangles are red (the clinical Ki-67 breast-cancer high-risk cutoff, 20%, 12/60 exactly). Unlike the 8 core rules, this one was added later (Section 4.1) via its own dedicated generator (`gen_proportion_threshold.py`) rather than a `configs/rules/` YAML file, so it has no config-driven train/val/test split – VQVAE, SlotAttn-Tr, and Oracle were nonetheless trained on data from this generator. Its layout also differs: left block, 16 comfortable-margin TRUE scenes (`easy_positive_gate`, 30/60 red); middle block, 16 tightest-boundary FALSE scenes (`boundary_false_off_by_one`, 11/60 red, one below threshold); right block, 16 ordinary FALSE scenes (0–5 red triangles, not a near-miss) – no separate ordinary-FALSE category beyond this.

A.21 MODEL OVERVIEW

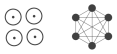
One card per model in the roster, for reference. Parameter counts are computed directly from each model's own frozen checkpoint (sum of `state_dict` tensor sizes, from each model's seed-1 checkpoint on `at-least-one-touching-pair`; any seed/rule is representative since model size does not vary by either), not estimated. Training hardware (2× Tesla V100-PCIE-32GB throughout) is taken from this project's own saved GPU driver logs; per-model average training time is not recorded in any saved log (only per-checkpoint filesystem timestamps exist, which conflate multiple concurrent runs on shared hardware) and is omitted rather than estimated. For VQVAE, SlotAttn-Tr, and Oracle, "mean behavioural fit" is standard-distribution program-vs-decision agreement, averaged over the broader 20-rule behavioural-recovery survey (Appendix A.7) – not the 8-rule boundary-adversarial testbed the rest of the paper centers on. It is a measure of in-sample fit, not of boundary robustness, so a rule can score well here (e.g. `touching-count-matches-red-left`'s 0.904–0.923) while still collapsing at its own decision boundary (Section 4.2) – the two numbers answer different questions and are not in tension. DiffusionClf has no recovered program to compare against (its own card, below), so its figure is a different quantity, labeled accordingly: mean classification accuracy against ground truth over its own 9-rule sweep (Appendix A.16).



VQVAE

VQ-VAE tokenizer + relational Transformer

- **Perception**: patch tokens [3].
- **Relational**: self-attn. Transformer (shared w/ SlotAttn-Tr, Oracle).
- **Parameters**: 2,853,188.
- **Training**: standard per-rule regime, from scratch, 3 seeds.
- **Mean behavioural fit**: 0.800 (20-rule survey).



SlotAttn-Tr

Slot Attention encoder + same relational Transformer

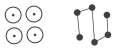
- **Perception**: object-centric slots [1].
- **Relational**: self-attn. Transformer (shared w/ VQVAE, Oracle).
- **Parameters**: 2,815,076.
- **Training**: standard per-rule regime, from scratch, 3 seeds.
- **Mean behavioural fit**: 0.738 (20-rule survey).



Oracle

Perception-free upper bound: true object list given directly

- **Perception**: none – rules out pixel/segmentation noise.
- **Relational**: self-attn. Transformer (shared w/ VQVAE, SlotAttn-Tr).
- **Parameters**: 2,670,916.
- **Training**: standard per-rule regime, from scratch, 3 seeds.
- **Mean behavioural fit**: 0.940 (20-rule survey) – highest, as expected of an upper bound.



SlotAttn-RN

Slot Attention frontend + Relation Network Stage 2

- **Perception**: object-centric slots (reused from SlotAttn-Tr).
- **Relational**: Relation Network, no self-attention [2] – enables tractable LRP/CRP (App. A.19).
- **Parameters**: 474,785 – far fewer (shared pairwise MLP).
- **Training**: standard per-rule regime, from scratch, 3 seeds, 9 of the 10 rules (no checkpoint on the published 60-triangle `red-triangle-proportion-ge-20pct` construction; App. A.15's separate 120-triangle construction has its own), classification-only – no recovered program, so no `ground_truth_fidelity`.
- **Mean behavioural fit**: n/a – classification-only.



DiffusionClf

Pixel-space conditional DDPM, used as a classifier

- **Perception**: none separated out – single UNet on raw pixels, no tokenized front end.
- **Relational**: none explicit – must emerge from the denoising objective alone.
- **Parameters**: 3,912,579.
- **Training**: from scratch per rule, 3 seeds; classified via the noise-prediction-error trick (App. A.16).
- **Mean classification accuracy**: 0.698 across 9 rules, below the 0.95 validity gate on 8/9 (App. A.16).