WHEN STRONGER BASELINES MISLEAD: ADMISSIBILITY AUDITING FOR REPRESENTATION-LEVEL CLAIMS

Anonymous authors
Paper under double-blind review

ABSTRACT

A baseline is evidence against an alternative explanation only if it is *invariant* to the claimed property, so a stronger baseline need not be a stronger control. On AI Feynman a zero-parameter bag of variable tokens scores $\mathbf{1.000}$ against a Tree-LSTM's 0.972, so that encoder's score is no evidence about invariance to variable identity.

A baseline comparison estimates *relative performance*; *admissibility auditing* estimates the *ceiling* of a *declared* family $\mathcal{F}$: the best score *any* control strong on the task yet blind to the claimed property can reach, *trained* readouts included, at three cue levels S1–S3. That direction of comparison is forced: $M(h) > \sup_{g \in \mathcal{F}} M(g)$; *which* family we declare is the choice. A pass rules out the declared alternatives, and *no* admissible family can turn it into a certificate: **the audit falsifies by design**. The levels name *cues*, not algebra: the audit ports to language and to code, where *our own* inversion fails to replicate.

Against a shape-matched twin (same tokens and operators, different *arrangement*), **every arrangement-invariant representation, declared or not, is pinned at exactly 0.500 by proof**, and a Tree-LSTM reaches $\mathbf{0.994}$; on unseen classes it scores $\mathbf{0.894}$ against 0.676 for the strongest order-blind control an adversarial search finds. **Four of the audited results are other people's**, ten in all, eight of them already published: a released 80M-parameter integration encoder scores 0.872 where a zero-parameter operator/arity bag scores $\mathbf{0.941}$ (an S2 task correctly identified, not a defect), while a 14-corpus leaderboard is *upheld*. So "strongest baseline" is not an evidential principle: the ceiling is *non-monotone* in resolution, and widening the family can only cost a pass.

What survives every control is narrower than "compositional generalization". A Tree-LSTM trained on *single* rewrites identifies held-out *compositions* at depth 8 on `poly8` (0.736) and clears a composition split designed by others, so it *passes the $\mathcal{F}_3$ audit*, as *composition-of-known-transformations* generalization, scoped to $K \geq 200$ here. That is not compositional reasoning: four primitives are not a library. **The audit is the deliverable; the findings are its test.**

1 INTRODUCTION

Take the claim that a Tree-LSTM recognises algebraic *form* rather than which *variables* appear. On AI Feynman a zero-parameter bag of variable tokens scores $\mathbf{1.000}$ where that encoder scores 0.972 (Figure 2, rung S1): the better predictor of the benchmark label is worthless as evidence about variable-identity invariance for precisely the reason it wins, being a function of the variables alone. So admissibility must be a *test* a comparator passes, never a judgement of its strength. **The criterion, in one sentence: a learned representation h is evidence for a property P only if it outperforms the *best* representation in a *declared* family $\mathcal{F}$ invariant to P, *trained* readouts included, under the same protocol and score M; declared, never complete (§3.3).**

$$\text{evidence about } P \text{ requires } \; M(h) > \sup_{g \in \mathcal{F}} M(g), \; \text{not merely } \; M(h) > M(g).$$

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

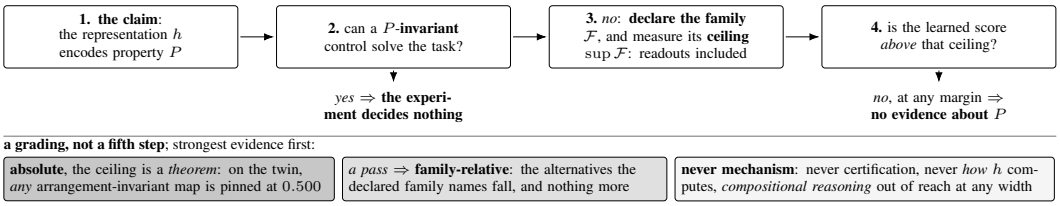

Figure 1: **The audit in one picture: steps 2 and 4 end the inference, and only step 2 before any learned score is read**; the band below grades what a pass means. **S1–S3** name *cues*; $\mathcal{F}_3$ is the controls reading *only* cues up to S3, and *coverage* is a separate axis, not a fourth level (§3.2).

So this is a paper about evaluation, not about encoders: the contribution is that protocol (we call it *admissibility auditing*) and symbolic-expression encoders are the *case study* where it revises published conclusions (Table 10), language and code two ports of the same code.

Where the flaw actually lives. A zero-parameter classifier over 25 corpora, 15 of them EQNET variants (§4.1), reports separability at each level. **S1 is clean almost everywhere**: the canonical published family (Allamanis et al., 2017; Gangwar & Kani, 2023; Zheng et al., 2025) defines classes over a shared variable pool; and **S2 is where the exposure is**, at 66–100% of class *pairs*: a mismatch between cue and claim, not a cheat. AI Feynman (Udrescu & Tegmark, 2020) is the severe case, not the typical one: a constructed counterexample, not evidence of prevalence.

Three objects, and only the third is evidence (Figure 1), one question each: **(1)** a *baseline* ⇒ did the model score higher? **(2)** an *admissible control* ⇒ could something blind to P have scored that? **(3)** the *family-relative admissible ceiling* $\sup\mathcal{F}$ ⇒ could the best such control have, over trained readouts as well as feature maps? **Only the third is the audit statistic**, and *no* published evaluation reports it: one cell, $+0.874$ or $+0.22$ by control choice (six objects are the critical path: Figure 1; Table 2, every claim and its verdict; Table 1, the axes no prior device attains; Table 10; §4.2's twin; and §3.3). What it refutes is an *evidential* interpretation, not the hypothesis itself.

One example fixes the three levels. The class of $x{+}y$ contains $y{+}x$ and $(x{+}0){+}y$: a variable bag (**S1**) separates nothing, while an operator/arity bag (**S2**) separates them perfectly, $\{+\}$ against $\{+,+\}$, without parsing either form. **That is the trap**: beating S2 is not evidence of composition when which operators appear is itself part of the structure being claimed. **S3** adds local shape and is order-blind by construction: $x{+}y$ and $y{+}x$ are identical to every ϕ_d.

The sharpest form of the test: a ceiling that is a theorem. Hold every lower-level cue fixed: pair each held-out form with a *shape-matched twin*, the same variable multiset and the same operator multiset in a *different arrangement*, and the ceiling stops being an empirical search. **Every representation invariant to arrangement, *not* only the declared family's members, is pinned at exactly 0.500, chance, *by proof*, a trained readout over one included at any capacity**, while a Tree-LSTM reaches $\mathbf{0.994}$ (§4.2; Prop. 4; rung 4 of Figure 2).

Applying the criterion changes conclusions, in three places. Every part of its distance from "use an invariant control" is *measured*: membership tested per instance, the statistic the declared family's *ceiling*, not any member, and a supremum over *trained readouts* too (Prop. 1; §3.3). **(i) *Empirical***, an audit of published claims: the same screens revise ten claims across *three* modalities (eight already-published results, four of them other people's), in *both* directions: one broken, one **upheld**, three narrowed, one re-scoped to S2, one restated, one settled below S3, two withdrawn against ourselves (§4.4, Table 10). Predictions were registered in the runners before the runs; four failed, and all four are printed. **(ii) A *consequence***: an evaluation protocol is itself a hypothesis about what generalization means; two protocols that look equivalent license different readings of *one* set of weights (§4.5), and within one the ceiling is measured non-monotone in resolution (Corollary 2), while no declarable family manufactures a pass (Corollary 3), a bound we try to break three ways: by hand, by adversarial search, and by re-scoring that search's winner, **frozen**, on *partitions it never saw*. **(iii) *Positive***, a demonstration, and its exact scope: what survives every control is narrower than "compositional generalization"; on `poly8` and two further EQNET corpora, one a *different algebra*, a Tree-LSTM *passes the* $\mathcal{F}_3$ *audit* up to unseen compositions of *known* rewrites, novel arrangement far cheaper than novel inventory, and on a composition split designed by others (§4.4).

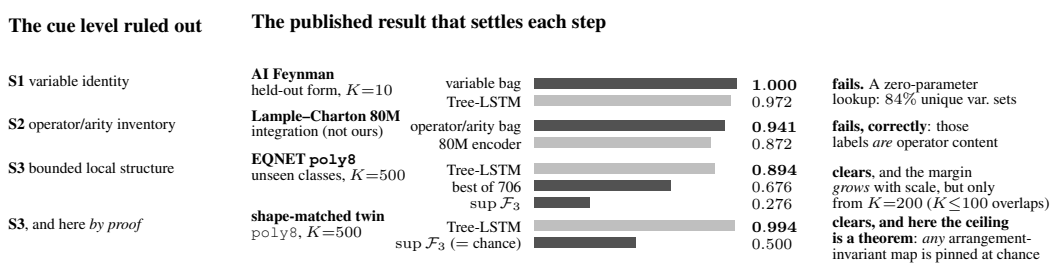

Figure 2: **The twin is the one rung whose ceiling is known by proof**: *any* arrangement-invariant representation is pinned at chance, and tree-edit distance, the strongest order-sensitive *non*-member, reaches only 0.723. Each rung settles one step of Figure 1's chain, at the level named on the left; dark bars are non-learned controls, light bars trained models, and the unit of inference is the equivalence class on `poly8` and the individual expression above it. **Coverage is not a rung** but one of the five novelty axes costed in §4.5, and not a monotone gate; its ordering fails on `boolean8`.

2 Related Work

Neural embedding of symbolic-expression equivalence classes. The line this paper audits begins with *neural equivalence networks* (Allamanis et al., 2017), which place semantically equivalent but syntactically dissimilar expressions together and evaluate on held-out classes (UnseenEqClass); Gangwar & Kani (2023) and Zheng et al. (2025) obtain equivalent pairs differently. All three define classes over a *shared* variable pool (Table 23).

Shortcut learning, annotation artifacts, and control tasks. Geirhos et al. (2020) frame *shortcut learning* as a pervasive failure mode; in NLI, hypothesis-only baselines matched full models (Gururangan et al., 2018; Poliak et al., 2018), and our zero-parameter bag is the symbolic analogue (Table 22). Table 1 places the rest; two are near enough to state here: a *matched* control (Elazar et al., 2021) is invariant by construction along the one dimension it matches, not over a declared family, and *counterfactual evaluation* (contrast sets; Gardner et al., 2020) perturbs the input and re-tests the model, where we fix the input and constrain the *comparator*. *Control tasks* (Hewitt & Liang, 2019) interpret probing accuracy only against a randomized baseline; ours is the *zero-parameter* bag, which the per-instance test admits, an *untrained* encoder being a secondary diagnostic that fails it (§3.2). Behavioural *invariance tests* (Ribeiro et al., 2020) ask invariance of the *model*'s prediction; renaming-invariance has a code analogue in Type-2 clone detection (Svajlenko et al., 2014; Zhang et al., 2019; Wang et al., 2020).

What is new, given all of that. Prior devices test a guessed comparator, perturbation, or *adversarial* search in isolation; **none estimates** $\sup_{g\in\mathcal{F}} M(g)$**, a declared family's *ceiling* over *trained* readouts** (§3.3). **Table 2's row 1 is the cost of guessing wrong**: 0.972 read as structure, where an admissible map reaches 1.000. **The question that becomes answerable is not whether a model exploits a shortcut, but whether the experiment could have told**: theory derives shortcut bias from training data (Lippl & Stachenfeld, 2025); the *ceiling itself* is our inferential object.

Held-out construction, contamination, and the models we use. SCAN (Lake & Baroni, 2018) and COGS (Kim & Linzen, 2020) established that *held-out construction* determines observed difficulty; our matched contrast (§4.5) is the symbolic analogue, and **we audit SCAN itself, where the ceiling is *monotone*** (§4.4, Appendix AX); the contamination literature (Jacovi et al., 2023; Zhou et al., 2023) recommends shortcut-breaking held-out sets. Our encoders are standard (Tai et al., 2015; Xu et al., 2019; Vaswani et al., 2017; Lample & Charton, 2020; Schroff et al., 2015; Wen et al., 2016; Snell et al., 2017), as is embedding propositional formulae by satisfaction supervision (Xie et al., 2019); symbolic-regression systems (Udrescu & Tegmark, 2020; Biggio et al., 2021; Cranmer et al., 2019) supply two benchmarks we audit, and all three architectures leak variable identity under renaming, which locates the mechanism in the benchmark, not the encoder.

Table 1: **Takeaway: "strongest baseline" clears *nothing***, × in all eight columns: strength is not blindness, which is what the three **bold** columns test, and **no prior row attains any of them**: that conjunction is what is new. **The last column is × on every row, ours included** (§3.3).

Prior device	benchmark solvable?	property-relative?	searches a class?	**comparator invariance?**	**family ceiling?**	**readout closure?**	falsifies only?	certifies mechanism?
shortcut baseline	✓	×	×	×	×	×	✓	×
control task	partly	partly	×	×	×	×	✓	×
matched control	partly	✓	×	**partly**	×	×	✓	×
strongest baseline	×	×	×	×	×	×	×	×
adversarial baseline	✓	×	✓	×	×	×	✓	×
invariance test	×	✓	×	partly	×	×	✓	×
counterfactual eval.	✓	✓	×	×	×	×	✓	×
held-out split	×	×	×	×	×	×	×	×
learned inv. control	✓	✓	partly	partly	×	partly	✓	×
admissibility auditing	✓	✓	✓	✓	✓	✓	✓	×

3 METHODOLOGY

3.1 DATA, EQUIVALENCE CLASSES, AND ENCODERS

We work over binary expression trees over {+, -, *, /, sin, cos, exp, log, sqrt}. The cautionary corpus is AI Feynman (Udrescu & Tegmark, 2020): 10 equations, each drawing a distinct subset of a 16-symbol pool, the criterion under audit. Equivalence classes come solely from a 13-schema paraphrase library; domain guards and the rejected numerical alternative are in Appendices A, Q.

A disclosure that bounds everything below. Classes *we* generate hold $\approx$3 members, each positive a *single-rewrite* paraphrase with near-total token overlap: the easiest possible positives. The argument is about evidence *validity*, not benchmark difficulty: an easy positive a P-blind control also solves is exactly the case the audit is for. It bounds our own positives too: a pass establishes recognition of single-rewrite equivalence. For the uneven external EQNET corpora every null and attainable maximum comes from the measured size distribution (Appendix Z).

Three encoders, all 128-dimensional: a Tree-LSTM (Tai et al., 2015), a GIN (Xu et al., 2019) over the expression DAG and a Transformer (Vaswani et al., 2017) over a bracketed linearisation, all trained with satisfaction cross-entropy plus semi-hard triplet loss (Schroff et al., 2015) at margin α=0.2.

3.2 THREE LEVELS OF REPRESENTATIONAL SHORTCUT, AND A COVERAGE AXIS

"Structure" is not one thing, so we fix **three levels of shortcut**; every claim names its level.

Level	The control reads	It cannot tell apart	**If it wins**, the result is explained by
S1 variable identity	*which* symbols appear: a variable bag	$x+y$ from $x\cdot y$	**notational identity** alone
S2 operator/arity inventory	which operators, over how many variables: op bag	$(x+y)\cdot z$ from $x\cdot(y+z)$	the **operator inventory**
S3 order, depth, local shape	adjacency, depth profile, parent–child statistics, subtree sizes: ϕ_d, WL_h	$x+y$ from $y+x$	**local structural statistics**
protocols disagree	*not a control: two metrics, one set of weights*		the **metric**, not the representation

S2 is the level to watch, and operator identity is *legitimate* information: x_0/x_1 is not $x_0 \cdot x_1$, so a corpus separable at S2 is mis-described, not broken, but reading a multiset is not parsing.

Terms.

cue information a form carries; $\mathcal{F}_t$: the controls reading levels $\leq$St only. **twin**: a non-equivalent form matched on every lower-level cue. **coverage** (axis C): was the held-out form's rewrite in

Table 2: **Takeaway: every claim, the strongest admissible control we found and the verdict on one page, and the audit moved the interpretation *both* ways.** The unit of inference is the equivalence class on our corpora and the individual expression on the two external ones. Reported evidence in the first block was in every case a comparison against the strongest baseline then available; below the second rule the audit is turned on our own family, readout and portability claims (§4.1, Appendix F). Each row is an **audit certificate**; the columns are its parts (Def. 1).

Claim audited	Level	Strongest admissible control	Reported	Verdict
Other people's published claims, audited prospectively by the criterion of §3.2				
AI Feynman variable invariance	S1	variable bag 1.000	0.972	broken
Lample–Charton 80M integration	S2	op/arity bag 0.941	0.872	S2 only
score$_5$ leaderboard, 14 corpora	S1, S2	*none* reaches it	—	upheld
StructEmb ablation, 14 corpora	S1, S2	an untrained encoder, on 5	—	narrowed
Our own claims, audited by the same criterion				
Our poly8, unseen classes	S3	an adversarial search, 0.676	0.894	pass, and only $K \geq 200$
on the shape-matched twin	S3	*any* arrangement-blind map, 0.500 *by proof*	**0.994**	pass; the ceiling is a theorem
Composition of known rewrites, d=8	S3, C	non-learned 0.100	0.736	supported
on a split we did *not* design	S3, C	sup $\mathcal{F}_3$ 0.0005	0.987	supported
Our inversion, off symbolic math	S3	SCAN's ceiling 0.115 $<$ its ϕ_∞	—	bounded
Our verdicts under a *wider* family	S3	the union, and all 8191 sub-families	$\leq$+0.038	no *pass* moves
under *every* trained readout	S3	the readout-free bound	6/8 cells	narrowed at K=50
Our NeSymReS portability claim	S1	one global permutation	+0.024	withdrawn
General compositional reasoning	—	—	—	out of reach in principle

the training library? **family-relative**: admissible by a declared family's per-instance test, never by proof of global invariance; a *ceiling*, though, can be (Prop. 4). $\mathbf{sup}\,\boldsymbol{\mathcal{F}_t}$: §1's audit statistic, at level t.

S3, made precise: a bounded feature family ϕ_d. For a depth bound d, let $\phi_d(T)$ be the multiset, over every node of the variable-anonymised tree T, of the canonical string of the subtree rooted there truncated to depth d, children unordered, **order-blind by construction**. Its order-sensitive completion ϕ_∞ (whole subtrees, children in order) is an *extension*, not a member (Appendix AC).

Theorem 1 (A simple formal justification for why admissibility ceilings are the appropriate statistic)**.** *Fix a metric M and protocol and a property P; let $\mathcal{A}_P$ be* every *representation invariant to P and $s = \sup\{M(g) : g \in \mathcal{A}_P\}$.* (i) *If $M(h) \leq s$, two regimes (h reading P, and h reading a P-blind cue) induce the* same *pair $(M(h), M(g))$ for* every *comparator g (construction in Appendix AN).* (ii) *Hence $M(h) > s$ is necessary as well as sufficient to remove the ambiguity.* (iii) *A g that is* not *P-invariant bounds s at no value of $M(g)$, so comparing h against it decides nothing about P, in either direction.*

Corollary 1 (Invariance falsification)**.** *Let g be any representation invariant to P: $g(T) = g(T')$ whenever T, T' differ only in P. If $M(g) \geq M(h)$, $M(h)$ is not evidence that h encodes P.*

The asymmetry, the whole reason the criterion only ever falsifies: $M(g) < M(h)$ shows only that h is not P-blind, and each level's control keeps that level's cue and discards everything above it, rather than ablating the model, so it falsifies a claim about structure above its level, not about the cue itself. **Three of these numbered results are not definitional**: closure puts a trained readout *inside* the family (Prop. 1), so every ceiling here is a *result*, never a stipulation; the ceiling is then non-monotone in resolution (Cor. 2); and **no admissible family certifies** (Prop. 3: *unreachable*, not unbuilt); not one of the three a property of suprema. Theorem 1 itself is billed as *scoping*.

Definition 1 (Passing the $\mathcal{F}$-relative admissibility audit through level ST, and coverage-gated)**.** *Let $\mathcal{F}_t$ be a set of representations each determined by the cues at levels $\leq t$ alone,* and hence invariant to the structure those cues do not determine*. An evaluation of h under M* passes the $\mathcal{F}$-relative admissibility audit through level ST*, $T \leq 3$, if for every $t \leq T$, $M(h)$ exceeds* $\sup\{M(g) : g \in \mathcal{F}_t\}$ *under the same protocol with class-level intervals excluding zero. It is separately* coverage-gated *(axis C) if the held-out forms lie outside the training rewrite library. Ours, always: $\mathcal{F}_1$ variable bags, $\mathcal{F}_2$ operator/arity bags, $\mathcal{F}_3 = \{\phi_d\}_{d\in\{1,2,4,8\}} \cup \{\mathrm{WL}_h\}_{h\in\{1,2,3\}}$, the WL radii conventional* 1–3 *rather than derived.* ***The depth ladder is not a hyperparameter: it*** **terminates***; poly8's deepest tree is* 4*, so $\phi_{d\geq 4}$* is *the complete order-blind fingerprint and no finer member exists (§4.2).*

What is and is not guaranteed. Four kinds, kept apart. *(i) Necessary*, by Theorem 1: the direction of comparison and the supremum as instrument, nothing about which class. *(ii) Methodological, and ours*: $\mathcal{F}_t$ is declared, a testable stand-in for the $\mathcal{A}_P$ nobody can enumerate. *(iii) Empirical*: passing

establishes not compositional reasoning, only that the learned representation exceeds *every member* of a prespecified family under the protocol; **it does not establish that $\mathcal{F}_3$ contains all P-invariant explanations**; we say it passes the $\mathcal{F}_3$ audit, so the family travels in the name; *(iv) Proved, not open*: $\mathcal{F}_3$ is *bounded* and composition is not, so completeness is unreachable (Cor. 3; Props. 2, 3).

Membership in $\mathcal{F}$ is *tested* rather than assumed. The test is the twin, per instance: a member is pinned at exactly 0.500; *every* invariant map is, by proof (Prop. 4), so a baseline reading arrangement cannot enter however well it scores. Passing admits g operationally: admissibility here is family-relative. **The size of that gap is measured, not conceded**: over 17 cells and six corpora the verdict is unchanged (Appendix AA). The enumeration is not the family's boundary: a reader can admit a candidate we never considered by the same test. ϕ_∞, tree-edit distance and an untrained encoder fail it and are reported as extensions outside the criterion, which costs nothing (Prop. 1).

Proposition 1 (The admissibility ceiling)**.** *Let $\mathcal{F}_t$ be as in Definition 1, so that membership* requires *invariance to the structure that level t's cues do not determine. Refining any $g \in \mathcal{F}_t$ either preserves that invariance (leaving g in the family, where the supremum already dominates it) or destroys it, forfeiting membership. $\mathcal{F}_t$* ***is moreover closed under post-composition****: $f \circ g$ is determined by exactly the cues g is, so $f \circ g \in \mathcal{F}_t$ for* every f, including a trained f, *since fitting changes f and not what g reads. Hence the supremum, the family-relative admissible ceiling, is the audit statistic, taken over readouts as well as feature maps, and no member can stand in for it.*

Corollary 2 (A family is not audited by its strongest member)**.** *Prop. 1 does not* locate *the supremum, and it falls above any one member's point estimate: $M(\phi_d)$ is measured non-monotone in d, and* fitting *the readout raises it. So an S3 pass follows from no single member* and no single readout.

Corollary 3 (Which family we declared cannot manufacture a pass)**.** *The ceiling is a supremum, hence monotone under inclusion, so clearing $\mathcal{F}_t$ clears* every $\mathcal{F}' \subseteq \mathcal{F}_t$: ***leaving a descriptor out is strictly against our own interest****, and which family we declared bears on how informative a* failure *is, never on whether a* pass *is sound (§4.2).*

3.3 A Baseline's Strength Bounds Nothing; the Family's Ceiling Does

The property claimed here is sensitivity to algebraic arrangement, and a control is admissible only if it can score well *without reading* it, which a *twin* tests per instance. At `poly8` $K{=}500$, the cell §4.2 prints, the Laplacian spectrum scores 0.020; $\mathcal{F}_3$'s best single member under one fixed readout, 0.276; that same member with the readout *fitted*, 0.284; the bound over *every* readout, nothing trained, 0.296; and the full-token bag, order-blind too but reading the variable identity ϕ_d anonymises away, 0.517; 0.676 once such bags may be *composed* (§4.2). **So against the encoder's** 0.894 **the apparent margin is** $+0.874$**,** $+0.618$**,** $+0.598$**,** $+0.377$ **or** $+0.22$ **according to which control is called "the invariant baseline"**, which is why the statistic must be a supremum and not a member. Nor is the difference theoretical. Fitting the readout raised the ceiling at *every* cell we measured, by up to $+0.107$, and closing the family over readouts leaves one of our own passes unproved by 0.0027 at `boolean8` $K{=}50$ (0.8706 against a bound of 0.8733, where its strongest member, 0.732, would have shown $+0.139$; Appendix BA). And each $\mathcal{F}_t$ is declared, as is the level hierarchy above it, so what the audit can measure bounds Theorem 1's s from below, never above:

$$\sup_{g \in \mathcal{F}_t} M(g) \;\leq\; s \;=\; \sup_{g \in \mathcal{A}_P} M(g)$$

Hence a **graded verdict**: a *failure* is conclusive; a *pass*, only family-relative (Cor. 3); at a *twin*, absolute (Prop. 4). **That grading is the deliverable**: Table 2 lists what each rules out.

One standard practice per row, and what admissibility requires in its place; measured, not judged, and four times the measurement reversed a conclusion of *ours* (Appendix U):

a *strong* baseline ⇒ **admissible**: a non-P-invariant comparator bounds nothing at *any* score (Thm 1(iii))
a *control task* ⇒ invariance **tested per instance**, by a twin, not assumed (Appendix AA)
a matched control ⇒ a **declared family** and its *ceiling*, which no member replaces (Prop. 1, Cor. 2)
a *good score* ⇒ read only above that ceiling, **falsification tools, not certification tools**
a trained model ⇒ **admissibility is a property of the comparator's input pipeline, not of its strength**: a learned readout over an order-blind map is a member *at any capacity* (Prop. 1)
"*out-of-distribution*" ⇒ **five separately costed axes**: novel *arrangement* far cheaper than novel *inventory* (§4.5)

Table 1 runs the first five across nine prior devices; **the nearest is a *learned* invariant control**, invariant *by construction* rather than tested per instance, and one comparator rather than a family and its ceiling.

3.4 Held-Out-Form Protocol and Leakage Controls

We withhold one equivalent form per equation and test exclusively on it; the metric is *diagonal-correctness*, the fraction of held-out embeddings whose nearest training-equation centroid matches the true equation, chance $1/K$. S1 is isolated by *renamed-in-range*, remapping held-out variables inside the trained range to avoid the untrained-slot confound, S2 by *operator scrambling* and *arity control*. **Primary evidence is the *pair*** (plain, renamed), with both non-learned baselines.

Audit protocol. Figure 5 draws Definition 1 as a procedure, subject to two requirements it cannot show: each control must be an isomorphism of the benchmark, and the evaluation must match what it is compared against. **Three uncertainties, kept apart**: seeds quantify training stochasticity only, and split construction and family *specification* are the other two, each bounded on its own (§3.3, Appendix AL).

4 Experiments

4.1 Only One Family Carries the S1 Flaw, and S2 Is an Entitlement

Over all 25 corpora (Table 23, Appendices Y, AF, AG) **the S1 flaw is real and confined to one published family**: only AI Feynman (Udrescu & Tegmark, 2020) is exposed, at 84%.

Two encoders we did not build. On 320 integrands from Lample & Charton (2020)'s own test distribution, their released 80M-parameter integration encoder scores 0.872 and a zero-parameter operator/arity bag $\mathbf{0.941}$, **an S2 task correctly identified, and evidence of nothing above it**, since the four family labels are *defined* by which operators appear (Appendix AR). And on NeSymReS's published 10M Set-Transformer it returns a *null* under a global column permutation, withdrawing a portability claim of our own that an earlier, weaker control had scored as a drop (Table 2, Appendix I).

4.2 The Shape-Matched Twin: A Ceiling Known by Proof

Here the ceiling is proved, not searched for. EQNET `poly8` is the survey's only corpus clean at both lower levels, so it is where §3.2's criterion can be run: 1102 classes over a shared variable pool $\{x_0, x_1, x_2\}$ (S1 immune), *zero* of them with a unique operator bag (S2 immune). Its expressions and equivalence classes are EQNET's released data, used unmodified and hash-pinned; what is ours is the composition protocol layered on it. Tree-LSTM, GIN and Transformer encoders are trained on K classes under SeenEqClass and UnseenEqClass (Appendices Z, AD).

Holding every lower-level cue fixed: a ceiling that is a theorem. Each held-out form is paired with a twin of identical token and operator multisets, scored by which the encoder places closer to the class centroid (chance 0.5). Twin members are *not* algebraically equivalent, so a high score measures *composition-sensitive structural discrimination*, not algebraic-equivalence generalization. Arrangement alone differs, so any score above chance is read off arrangement. At $K{=}500$ the trained Tree-LSTM reaches $\mathbf{0.994}$, every *arrangement-invariant* representation, declared or not, is pinned at exactly 0.500 by proof (Prop. 4), readouts included, at any capacity; and tree-edit distance, the one baseline that *can* see a sibling swap, reaches only 0.723, so ***no*** **admissible comparator narrows the** 0.206 $[0.172, 0.240]$ **the untrained encoder leaves** (Appendices AB, AA).

Trained encoders exceed every non-learned baseline on unseen classes. A margin of $+0.22$, 0.181 clear of the encoder's bootstrap lower bound, over the strongest non-learned baseline an *adversarial* search over 28 primitives finds: at $K{=}500$, Tree-LSTM unseen $= 0.894$ against $\mathbf{0.676}$, a ratio $1.3\times$, not a flattering $6.1\times$ against a variable-blind bag. The constructive claim is scoped to $K{\geq}200$ on this corpus and design: intervals *overlap* at $K{=}50$ and 100 and separate cleanly from

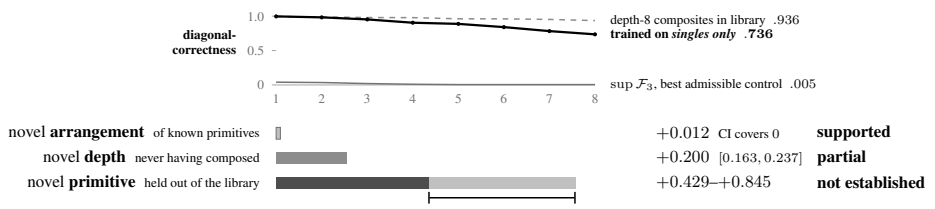

Figure 3: **Takeaway: diagonal-correctness (§3) decays gracefully with composition depth, and novel *arrangement* is nearly free where a novel *primitive* is fatal. Below**: paired drops at $d{=}8$ against that covered arm; the primitive span pools to $+0.664$; **the arrangement bar replicates on a second class set** (tags `r93`, `r94`; Appendices AU, AQ, AV; exposure $\times$ depth in full: AH).

$K{=}200$ on. That threshold is instrument resolution, not the effect's onset: unseen accuracy moves 0.05 over a $10\times$ K range ($0.843 \to 0.894$) while the class-level interval narrows $0.333 \to 0.066$.

The ceiling is attained by the *coarsest* member. The depth ladder terminates at $\phi_{d\geq 4}$ (§3.2). **And the finest is not the strongest**: the order-sensitive completion ϕ_∞ falls *below* the coarsest member (measured; Proposition 5, Appendix AC). It replicates on three corpora, two algebras, all five class partitions and an external encoder on deeper trees (Appendix AL), but not on SCAN (§4.4).

How much could the family we declared have mattered? Exactly this much. Widened by five descriptors we did not choose, the ceiling moves at 5 of 8 cells by at most $+0.038$, and all 8191 sub-families pass every cell the union passes (Table 2). So this measures robustness of the verdict to family specification; completeness is unreachable, never claimed (Corollary 3; Appendix AT). And widened by *search* rather than by hand: the 0.676 is *exploratory*, a hill-climb over weighted unions of those 28 primitives, each gated by the same per-instance twin test on the split we report, and it *asymptotes*: four optima in $[0.6523,\ 0.676]$, none closing the gap; **the *inferential* test: that winner frozen, re-scored on four partitions it never saw,** $+0.16$ **to** $+0.27$ (App. BB, Table 36). And over every trained readout, not just every feature map: a readout cannot separate what its input has merged, and *no* admissible map is finer than one, so one bound covers the family at any capacity, untrained; and it is cleared, with room, wherever the constructive claim is scoped (Appendix BA).

4.3 Trained on Singles; Tested on Depth-8 Chains It Never Saw

We train on single rewrite applications and test on held-out compositions equivalent to their anchor. Allowing each of poly8's four primitives twice at *distinct* sites reaches depth 8 over 200 classes, every held-out chain a token-delta-matched multiset in one of 2520 orders, 188 realised, *none* trained on. Trained on singles alone the Tree-LSTM identifies them at $\mathbf{0.736}$ at $d{=}8$ (chance 0.005; strongest non-learned 0.100) and a per-depth shape-matched twin holds $\mathbf{0.876}$ there, so the deep result is not the size contrast that grows with depth. **The evidence is sensitivity to global arrangement beyond the declared order-blind family**; calling that composition-of-known-transformations generalization is an interpretation that does not reach a primitive held out of the library, **or another encoder** (Appendices AU, AQ, AS; AK: a singles-trained Transformer fails the audit past $d{=}2$).

A different algebra: `boolean8` clears the criterion harder and breaks the coverage story. `boolean8` needs four new primitives, one of them non-local, and clears the criterion *more* decisively ($7.6\times$ against $5.0\times$) with the twin margin *widening* to $+0.296$ $[0.225, 0.364]$. And there the coverage mechanism breaks outright: `boolean8`'s coverage ladder is non-monotone; **the coverage-closure *mechanism* is polynomial- *and* encoder-specific** (Appendices AM, AK). And what the readout bound costs here: our $K{=}50$ pass is left unproved by 0.0027, and Appendix AO's untrained-encoder anomaly splits into a *readout* artifact there and a *proved* family property above it.

4.4 The Audit Ports; Our Own Inversion Does Not

An audit earns its keep where it *disagrees* with the conventional report, one row per modality, and in the natural-language row it disagrees with *us*. S1–S3 name cues, not algebra: everything above is the detailed case study, and a new modality needs only a parser and a level assignment; *portability, not general validation*, never prevalence.

Modality	Conventional report	What the audit licenses
symbolic math	a released 80M integration encoder scores 0.872 on its own test distribution	a zero-parameter operator/arity bag scores $\mathbf{0.941}$: **an S2 task correctly identified**, and evidence of nothing above it
natural language	a Tree-LSTM reaches $\mathbf{0.987}$ on SCAN `add_prim_jump`, where the grammar and the held-out set are theirs	it **passes**, against $\sup \mathcal{F}_3 = 0.0005$, *and their design puts the admissible bar on the floor*, so we quote no ratio: what clears it is composition of *one known* transformation onto one unseen primitive, still not compositional reasoning
code	S2 reaches 0.987 over 300 Type-2 clone classes we built from Python stdlib code	**Program representations: settled below S3**; *every* $\mathcal{F}_3$ member attains 0.990, exactly the maximum this corpus admits

The sharpest test of the whole procedure is one somebody else built: on Lake & Baroni (2018)'s split the prediction we registered before running held, and **it bounds our centrepiece**; there the ceiling is monotone, ϕ_∞ sits *above* the family, and §4.2's inversion replicates in *none* of 9 cells (Appendices AR, AX, AY, AW, AZ).

4.5 The Protocol Changes the Generalization Claim

Takeaway: an evaluation protocol is a hypothesis about what generalization means. On BOOLEAN8 ($K{=}190$), **one set of trained GIN weights reads as clearing S3 under one protocol and at chance under the other**: $\mathbf{.881}$ $[.850, .909]$ on UNSEENEQCLASS against a strongest-non-learned bound of .596, then $\mathbf{.511}{\pm}.026$ on the shape-matched twin against chance .500. The disagreement is the content: recovering a held-out form's class and telling it from a non-equivalent match are *different* properties. Neither GIN number reproduces on MPS; the .370 gap is $5.7\times$ the largest per-seed GIN drift, .065 (Appendix AJ). Which protocol a model clears is architectural: only the Tree-LSTM clears both; the Transformer's $[.588, .746]$ misses .596 (Appendix AO).

Novelty is not a scalar, so a split must declare which axis it varies. Changing only how the held-out form is *constructed* moves the apparent result further than any baseline in this paper does: "out-of-distribution" names *five separable axes*, free to fatal; the three paired at $d{=}8$ are drawn (Figure 3).

What the split makes novel	Measured cost	Verdict
arrangement of known primitives	$+0.012$, CI covers 0	free; also, unseen pair/triple $\leq +0.010$ (App. BC)
composition depth, never composed	$+0.200$ $[0.163, 0.237]$	graceful decay
schema, vs. a wrong-schema control	$+0.070$ $[0.021, 0.115]$	at the family, not the rewrite
inventory: library membership	$.965/.880 \rightarrow .832/.662$	plain / renamed; held-out tree *byte-identical*: largest move
primitive, held out of the library	$+0.429$–$+0.845$	not established; ≤ 5 inventory-safe anywhere (App. BD)

And the ranking is corpus-specific: this coverage ordering fails on `boolean8` (§4.3). So **name the axis, and declare the family against *it*** (Appendices AP, AQ, AV). **And this is no capacity failure**: our hardest recipe reaches only 0.480 out-of-library, against 0.992 in-library (App. H).

Scope of inference. One frozen pipeline throughout; inferential claims are restricted to the contrasts prespecified in §3.4, so every exploratory appendix test is *descriptive*. **The unit of inference is the equivalence class** on the corpora we built, and every interval above is a class-level bootstrap. Every pass falsifies a declared family's alternatives at a stated level and split, never certification, never architecture-independent (Appendix AK); scopings tied to one number stay beside it. **Where our own results break is the audit working**: the coverage mechanism does not port off polynomials (§4.3); a family-relative bar exists to expose that rather than absorb it. What we cannot yet show is a *reversal* on a non-symbolic benchmark built by others: the code corpus we settle is ours (**a refusal, at the corpus maximum**), and on SCAN the audit *narrows* (Table 10).

5 Conclusion

(1) The *direction* of comparison is forced; *which* family we declare is the choice, except at a twin: 0.500 **by proof** (Theorem 1, Prop. 4). **(2) "Strongest baseline" is not an evidential principle**: the ceiling is non-monotone in resolution, and widening the family can only cost a pass (Corollaries 2, 3). **(3) What survives every control is narrower than "compositional generalization"**: it *passes the $\mathcal{F}_3$ audit* as composition-of-known-transformations generalization, to depth 8 and on a split designed by others, not systematic compositional reasoning; scoped to $K{\geq}200$ on `poly8` and to the *polynomial* setting for the coverage mechanism (§4.5). **To audit your own claim** (Figures 1, 5): name the property, declare the invariant family, test each member *per instance*, estimate its ceiling, report the margin. The audit falsifies; it certifies nothing; Table 2 is the ledger.

ETHICS STATEMENT

This work involves no human subjects, no personally identifying information, and no new data collection: every corpus audited is already public (symbolic-expression benchmarks throughout, and, outside that domain, the published SCAN corpus (Lake & Baroni, 2018) and Type-2 clone classes built from a local Python standard library), and every model trained is a small encoder trained from scratch on those corpora. SCAN is fetched from its authors' repository under its own licence (BSD, Facebook Inc.) against a pinned SHA-256 and is *not* redistributed with this supplement. We see three ethical dimensions worth stating explicitly.

Auditing other researchers' published numbers. Section 4.1 computes non-learned floors under a live score_5 leaderboard. We report that the leaderboard is *robust*: no non-learned representation we built reaches EqNet or SemEmb on any of the 14 corpora; and we frame the result as a floor under a protocol, not as a defect in anyone's system. Where a corpus is weak (SIMPLEPOLY5) we name the corpus, not an author. The tier-1 finding concerns a *tokenization convention* inherited by a suite, not misconduct by the people who built the suite, and we say so wherever the finding appears.

Applying the standard to ourselves. The same audit has now changed *six* claims of our own; the four already-published numbers among them are the first four *Our…* rows of Table 10, and the other two are protocol-level: computed our own way, our floor appeared to beat a published ablation on 13 corpora rather than 5; both of our purpose-built "leakage-proof" suites proved solvable at tier 2; a float32 tie defect in our own non-learned baselines had understated them; the chance level of our headline metric was understated by up to $1.6\times$; and a $+0.250$ "alternate-order cost" we had published as a non-replication was an aggregation artifact, withdrawn in this revision, a correction that *removes* a caveat of ours rather than adding one; and, between review rounds, our own claim that the diagnostic detects notational sensitivity in a published 10M-parameter external encoder proved to rest on a control that permuted the held-out sets alone, so under a corpus-wide permutation the effect is a *null* and the claim is withdrawn (Appendix I). Every one is reported: the corrected numbers in the tables they belong to and the verdicts in Appendices T, P, AI and Z, with the full chronology of how each was found shipped as `REVISION_HISTORY.md` rather than deleted. We think an auditing paper that hid its own corrections would be self-refuting.

Compute and environmental cost. All experiments run on a single consumer GPU/accelerator; the largest single run (Table 26) is a few GPU-hours and the whole paper is well under 100. No large pretrained model is trained or fine-tuned.

We see no dual-use or misuse risk specific to this work: a tool that makes benchmark claims harder to overstate is defensive by construction.

REPRODUCIBILITY STATEMENT

Every quantitative claim in this paper carries a provenance tag (e.g. `r70`, `r76`) that names the script that produced it and the JSON log it was read from; Appendix K tabulates the tags of the AI Feynman and corpus-survey families, each EQNET run is tagged in the appendix subsection that reports it, and `REPRODUCE.md` lists every tag of both groups in one place. Appendix A documents the frozen pipeline configuration and names the rows that come from a separate run family. Code, the command-line auditor `audit-symbolic-benchmark`, all run logs, and the tier-1/tier-2 corpus statistics are included in the supplementary material.

The supplement also ships `verify_claims.py`, which makes 2619 assertions and exits non-zero if any fails, so corrupting a single logged value fails it. It *recomputes* the benchmark constructions the arguments depend on rather than reading them from a stored flag, and every check names the run tag it reads from and *fails* if that log is absent rather than skipping it. `REPRODUCE.md` indexes every run the verifier asserts against (all 87, a claim the verifier checks on itself), with each one's command, expected output and wall-clock runtime; six runs' logs have no wall-clock time and their cells say *not recorded* rather than an estimate. **That count has risen from 44, and 33 of the 43 it has gained are not new runs at all.** Thirty-two were a defect in the self-check: it derived its list by *reading its own source* for load sites with a literal string argument, so whole families addressed by a computed name (five class partitions $\times$ three scripts, four replicate runs) were asserted against while being invisible to the check that exists to prove they are indexed; it now records the stems as

they are loaded and expands the shorthands `REPRODUCE.md` uses to name a family. **The thirty-third is worse, and it is our own**: a log that had shipped since July was cited by no sentence of the body *and* carried no assertion at all, so the result answering the most obvious objection to §4.5's largest row was invisible twice over (Appendix H). Reading it in order to assert it found four wrong figures in the appendix that had been printing it, every one of them flattering; they are corrected there. **The other ten are new runs**, and counted as such, not folded into the defect: a composition-lattice run, two *invocations* of one library-expansion run (the second exists only to establish that the first's pre-registered stop rule was not an artifact of the first's own search budget, and both are indexed under one entry) and a 15-corpus library census; the remaining four, two encoders $\times$ two replicates, complete the one arm of the composition design this paper had recorded as never run, and are indexed under one entry for the same reason; the last two re-run the swap twin per class, one per corpus, for a class-level gap interval. This is bookkeeping, not evidence: it establishes that the numbers in the paper are the numbers the runs produced, and nothing about whether the audit is the right test.

Two caveats to state: first, the UNSEENEQCLASS protocol's no-leakage property rests on a single fixed shuffle seed that splits classes by index before training, so reproducing the split requires that seed; it is recorded in every log's provenance block, and Appendix AL re-runs the $K{=}500$ cell over five partitions, because no number of training seeds speaks to the choice of split. Second, training is stochastic and we report 3–25 seeds per run; the unit of inference throughout is the equivalence class (or equation), never the seed, and every interval bootstraps over that unit. Re-running a trained cell reproduces it to 0.0023 across environments and, on other arms, only to 0.062 across replicates (Appendices AL, AK), which bounds what the bit-for-bit claim in Appendix AJ covers. Appendix F records the revision history of numbers that changed during review.

STATEMENT ON THE USE OF AI ASSISTANTS

The research question, the level hierarchy, the falsification criterion at its centre, every decision about which claims the evidence supports, and every retraction reported in this paper are the author's. The author has read, checked, and takes full responsibility for the entire content, including all claims, numbers and citations, and for any errors that remain.

Where assistants were used. Large language model assistants (Anthropic Claude, via an agentic coding harness) were used substantially in implementing the experimental harness, the training-free auditor and the non-learned baselines; running experiments and reading their logs; drafting and revising the manuscript, including the wording of the sections here; and constructing the `verify_claims.py` checks. Portions of the pipeline used to organise the writing are themselves included in the supplement.

The failure mode of LLM-assisted scientific writing is confident text unsupported by the data, which is why the supplement ships an executable that asserts the prose against the logs as opposed to asking a reader to trust it.

REFERENCES

Miltiadis Allamanis, Pankajan Chanthirasegaran, Pushmeet Kohli, and Charles Sutton. Learning continuous semantic representations of symbolic expressions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 80–88. PMLR, 2017.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 936–945. PMLR, 2021.

Miles D Cranmer, Rui Xu, Peter Battaglia, and Shirley Ho. Learning symbolic physics with graph networks. *arXiv preprint arXiv:1909.05862*, 2019.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00359. URL `https://doi.org/10.1162/tacl_a_00359`.

Neeraj Gangwar and Nickvash Kani. Semantic representations of mathematical expressions in a continuous vector space. *Transactions on Machine Learning Research*, 2023.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1307–1323, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5075–5084, 2023.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning (ICML)*, 2018.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020.

Samuel Lippl and Kim Stachenfeld. When does compositional structure yield compositional generalization? a kernel theory. In *International Conference on Learning Representations (ICLR)*, 2025.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3: e103, 2017. doi: 10.7717/peerj-cs.103.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In **SEM*, 2018.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4902–4912, 2020.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12(77): 2539–2561, 2011.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Jeffrey Svajlenko, Judith F Islam, Iman Keivanloo, Chanchal K Roy, and Mohammad Mamun Mia. Towards a big data curated benchmark of inter-project code clones. In *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2014.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Wenhan Wang, Ge Li, Bo Ma, Xin Xia, and Zhi Jin. Detecting code clones with graph neural network and flow-augmented abstract syntax tree. In *IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2020.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, 2016.

Yaqi Xie, Ziwei Xu, Mohan S Kankanhalli, Kuldeep S Meel, and Harold Soh. Embedding symbolic knowledge into deep networks. In *Advances in Neural Information Processing Systems*, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Kaixuan Wang, and Xudong Liu. A novel neural source code representation based on abstract syntax tree. In *International Conference on Software Engineering (ICSE)*, 2019.

Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.

Hongbo Zheng, Suyuan Wang, Neeraj Gangwar, and Nickvash Kani. E-Gen: Leveraging E-Graphs to improve continuous representations of symbolic expressions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 11772–11788, 2025.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your LLM an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

APPENDIX

Reviewer map. One row per headline claim: where the body states it, the tagged run behind it, the appendix section that reports it, and the script in the supplement that reproduces it. Tags are the log stems `verify_claims.py` asserts against; `REPRODUCE.md` gives each one's full command line, expected output and runtime. **Two things this table is *not* pointing at.** First, the token-bag/Johnson–Lindenstrauss account of Appendix C is a *restricted-validity empirical* account and **not a theory**: it states its own validity domain, gives its own counterexample, and **no claim in the body depends on it**; it explains why an *untrained* encoder scores above chance, which is why the random-encoder row is a meaningful skyline, and nothing more. **That term, defined once:** a *skyline* is **one non-learned baseline's own score**, never the *ceiling*, which is a supremum over a whole declared family, trained readouts included (Prop. 1); Appendix O measures two skylines diverging 7× on identical forms, one of them far below the ceiling it is read as. Second, the tags `E3`, `E3b` and `E3m` in Appendices C–E are the AI Feynman held-out-form protocol variants and appear nowhere in the body: `E3` withholds an in-library paraphrase one rewrite step away (10 equations), `E3b` an out-of-library rewrite that changes operator structure, and `E3m` an in-library exp/log/power rewrite on the 4 eligible equations, in-library yet structurally divergent, which is what makes it the intermediate rung.

Claim (quoted from the body)	§	Run	App.	Code (supplement)
the S1 flaw is real and confined to one published family	4.1	`r57`	AG	`audit_symbolic_benchmark.py`
an S2 task correctly identified	4.1	`r90`	AR	`run_r90_external_audit.py`
withdrawing a portability claim of our own	4.1	`r62`	I	`run_r62_nesymres_global_permutation.py`
Trained encoders exceed every non-learned baseline on unseen classes	4.2	`r70`	AD	`run_r70_poly8_structural.py`
the finest is not the strongest	4.2	`r71`	AC	`run_r71_tier3_baselines.py`
robustness of the verdict to family specification	4.2	`r92`, `r98`	AT	`run_r92_family_stress.py`, `run_r98_family_robustness.py`
composition-sensitive structural discrimination	4.2	`r73`	AB	`run_r73_swap_twin.py`
composition-of-known-transformations	4.3	`r93`, `r91`	AU	`run_r93_deep_composition.py`, `run_r91_nonlocal_composition.py`
`boolean8`'s coverage ladder is non-monotone	4.3	`r84`	AM	`run_r81_composition_primitives.py --corpus boolean8`
the coverage-closure *mechanism* is polynomial- *and* encoder-specific	4.3	`r81_matrix_gnn`	AK	`run_r81_composition_primitives.py --arch gnn`
telling it from a non-equivalent match are *different* properties	4.5	`r86`	AO	`run_r73_swap_twin.py --corpus boolean8`
Program representations: settled below S3	4.4	`r95`, `r99`	AZ	`run_r95_code_audit.py`, `run_r99_code_structural.py`
it bounds our centrepiece	4.4	`r96`	AX	`run_r96_scan_audit.py`
the grammar and the held-out set are theirs	4.4	`r97`	AY	`run_r97_scan_composition.py`
moves the apparent result further than any baseline in this paper does	4.5	`r88`	AP	`run_r88_schema_transfer.py`
The size of that gap is measured, not conceded	3.2	`r74`	AA	`run_r74_probe_matrix.py`
admissibility is a property of the comparator's *input pipeline*, not of its strength	3.3	`r100`	BA	`run_r100_learned_invariant.py`
widened by *search* rather than by hand	4.2	`r101`	BB	`run_r101_adversarial_family.py`

How to read this appendix. What is being audited throughout is whether a representation-level claim is supported by its controls, not whether a model reasons compositionally, which this paper does not claim to settle (§3.3). **Six objects carry the argument**, and nothing else is needed to check it:

Figure 1 the audit as a procedure, and the two steps that can end the inference.
Table 2 every claim, the strongest admissible control *our search family found*, and the verdict, one *audit certificate* per row, whose parts are its columns (§3.3).
Table 1 what each prior device establishes, and the axes none of them attains.
Table 10 every published number the audit revised, and in which direction.
§4.2 the shape-matched twin, the one ceiling known by proof.

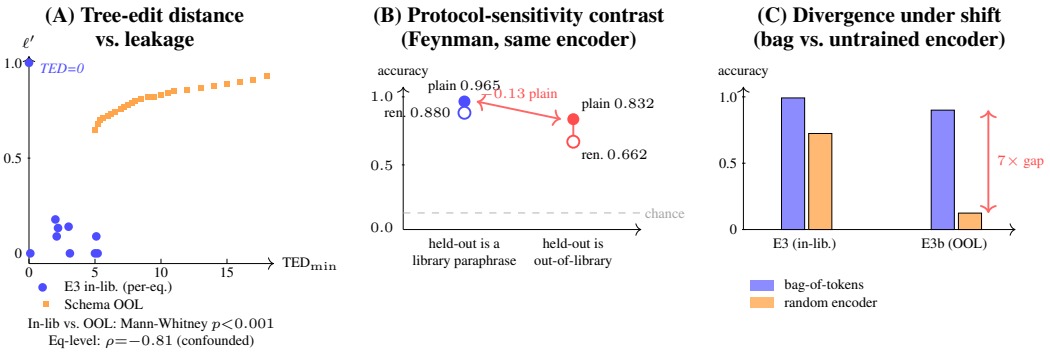

Figure 4: Three complementary views of variable-identity leakage. **(A)** Tree-edit-distance scatter: in-library forms show per-equation ℓ' (n=25 seeds each), schema-generated out-of-library forms show per-form ℓ'. One in-library equation (wavenumber, TED= 0) sits at $\ell'\approx 1$ because its held-out form is a training duplicate; it and the other TED= 0 form are excluded from the headline contrast. The two TED distributions are separated (Mann–Whitney $p<0.001$) but *overlap*; the equation-level correlation is $\rho=-0.81$ ($p=0.004$) while the residualized within-equation correlation is $\rho=+0.52$ ($p<0.001$): a Simpson's-paradox reversal we interpret in §4.5 without leaning on either sign. **(B)** The matched contrast of Table 21, reported as *(plain, renamed) pairs* rather than as an index: on one equation set of eight, one encoder and one training procedure, changing only how the held-out form is constructed drops plain accuracy by 0.13 and renamed accuracy by 0.22. **(C)** Divergence as a distribution-shift diagnostic: in-distribution, bag and untrained encoder track each other; under out-of-library shift they diverge 7× on identical forms with an identical centroid protocol.

§3.3 each standard practice, and the inference admissibility requires instead.

To have the ceiling itself checked rather than asserted: Appendix BA for the bound over every readout, Appendix AT for the widened family, and Appendix BB for the adversarial search over it.

The rest is long because every claim in the main text names the run it comes from, and each run brings its guards with it. It is organised as four things rather than one:

(i) A–K the ledger of every published claim the audit revised (Table 10, F), and the provenance, pipeline and defect detail behind it.
(ii) L–Z protocol specifications, the UNSEENEQCLASS k-NN design and its exact null (Z), the twin ladder (AB), and the ϕ_d family (AC).
(iii) AA–BD the experiments behind the main text, one self-contained section per run tag.
(iv) Proofs Appendix AN.

(i) is audit trail; (ii)–(iv) are the scientific content, so a reader who wants the science and not the bookkeeping can skip **A–K** *except* Table 10, which §1 cites and which is where the audit's effect on published conclusions is tabulated; a reader checking one number needs only (i) and the one section in (iii) that names its tag. The multiplicity is not exploratory search; it is this paper's own criterion applied recursively to its own results, each section one claim of ours put through the audit the body proposes, which is why several of them end by withdrawing or narrowing something we published.

The Reviewer map above indexes the 18 sections the body's headline claims rest on; the remainder are supporting runs, the depth-4 composition run that Appendix AU supersedes (AH), leave-one-primitive-out (AQ), five class partitions (AL), the non-local swap (AS), the order-diversity run that withdrew one of our own numbers (AV), the S2 audit of the shared-variable positive result (AE), and the port to code at S1–S2 (AW) that Appendix AZ finishes. A few stand alone as full tables or backup material to be read on their own, **Q** (domain guards and sampling), **U** (every claim against its strongest skyline), **V** (the score_5 floor in full), **AF** (the AI Feynman case study) and **AG** (EQNET under its own splits). The developmental chronology, how each correction was found and what it cost, is not here: it ships as `REVISION_HISTORY.md`, so that what is in the paper is the corrections' *outcomes* and the checks that now prevent them.

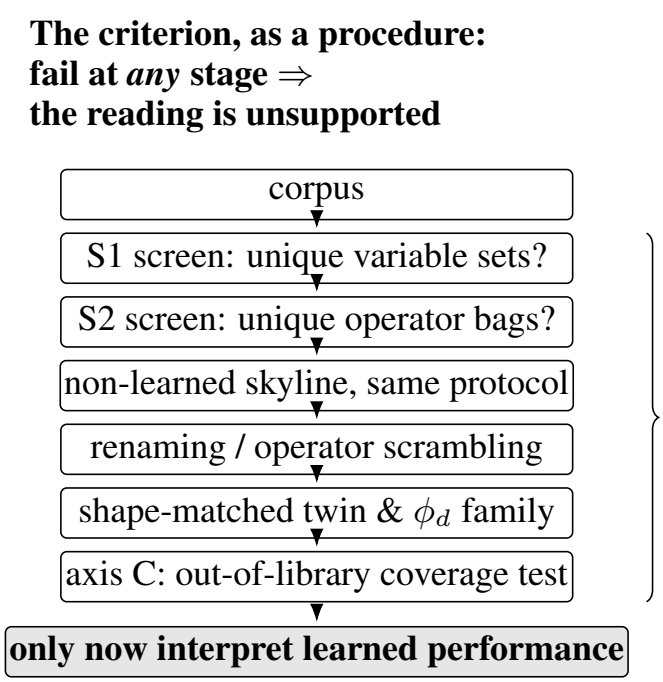

Figure 5: **Definition 1 as a procedure**, and what the released auditor implements. The first six stages are screens; the seventh is the only point at which a learned number is interpreted. **Coverage is drawn in sequence but it is neither a cue level nor a universal gate**: S1–S3 are representational invariances, whereas coverage is a property of the train/test split, one of the *five* independent axes of novelty costed in §4.5, and **measured non-monotone** on `boolean8`, where the coverage-closure mechanism does not hold (§4.2, Appendix AM).

A. Provenance, Pipeline, and Domain Guards

Provenance. The AI Feynman family, Table 3's corpus sizes and Table 22's held-out-form rows, derives from a single frozen pipeline configuration executed as one run family. The pipeline fixes: satisfaction-rate filter excluding expressions with satisfaction rate outside [0.1, 0.9], paraphrase-only equivalence class construction (numerical signatures were evaluated and rejected, as detailed below), extended exp/log/power rewrite rules, and early-stopped physics predictors (patience 10, maximum 100 epochs). No selective reporting or post-hoc parameter tuning was performed. Table 22 reports Tree-LSTM and Random at n=25 (one run family, fully independent seeds); GIN and Transformer remain at n=5 (separate earlier family). Table 17 (§N) derives from a second run family in which the alpha-renaming augmentation changes the paraphrase library, requiring fresh encoder training; its baseline and random encoder rows are independently re-evaluated under the same frozen pipeline and serve as within-family controls. The 10-equation protocol rows derive from tags r11 (E3), r13 (E3b), and r16 (E3m) within the Feynman diversity sweep (Appendix E). All comparisons are within-family unless noted.

Per-seed corpus sizes. The corpus is built by drawing 2000 base expressions per seed and expanding each into a size-≤ 3 equivalence class via up to $k = 2$ syntactic paraphrases; the build is fully deterministic given the seed. Table 3 reports the exact counts. The totals sit slightly above the nominal $2000 \times 3 = 6000$ because the paraphrase library occasionally returns fewer than two distinct rewrites, yielding a few size-2 classes and forcing an extra family to reach the target of 2000 base expressions. The reported "≈6000 expressions / ≈2000 classes" therefore denote these exact per-seed values, not rounded estimates. Each expression contributes n_{pos}=5 positive and n_{neg}=5 negative training samples (10 per expression).

Domain guards. Division by values $|y| < 10^{-6}$ clamps the denominator to $\text{sign}(y) \cdot 10^{-6}$; logarithm applies $\log(|x| + 10^{-6})$; square root uses $\sqrt{|x|}$; exponential clips arguments to $[-20, 20]$. The assignment sampler attempts up to $k = 64$ retries per expression; expressions for which no satisfying assignment is found within this budget are discarded. Variable assignments are sampled uniformly

Table 3: Exact per-seed corpus sizes (deterministic given the seed). "Size-3 / size-2" is the equivalence-class-size histogram.

Seed	Expressions	Classes	Size-3 / size-2 classes
0	6001	2001	1999 / 2
1	6001	2001	1999 / 2
2	6002	2001	2000 / 1
3	6002	2002	1998 / 4
4	6002	2001	2000 / 1

from $[-3, 3]$ for each of the 4 active variables per expression. Constants are drawn from a discrete bank $\{-2, -1, -0.5, 0.5, 1, 2, 3\}$, and thresholds are drawn from the same bank.

Satisfaction-rate histogram. The unfiltered corpus exhibits significant degeneracy: 46.8% of expressions have satisfaction rate outside [0.1, 0.9] over the $[-3, 3]^4$ sampling domain (33.1% always-false, 13.7% always-true). This motivates the filter that restricts the corpus to expressions with balanced satisfaction patterns. Numerical-signature hashing (256-bit SHA-256 of satisfaction patterns over a canonical assignment set) was evaluated as an equivalence-detection mechanism and rejected: pre-filter, 100% of randomly sampled pairs with identical signatures were false merges (2457 pairs tested, 0 verified by SymPy (Meurer et al., 2017)); post-filter, 13% remained false merges (100 pairs, 87 verified, 13 refuted by SymPy). Distinct algebraic expressions with identical satisfaction patterns over finite assignment sets are common in this domain. The final pipeline uses only syntactic paraphrases (commutativity, associativity, distributivity, additive and multiplicative identity, double negation, and exp/log/power rewrites: $a \cdot a \leftrightarrow \exp(2 \log a)$, $1/a \leftrightarrow \exp(-\log a)$, $\exp(\log a) \leftrightarrow a$) to define equivalence classes.

Quantization note. With 8 held-out forms per seed (or 10 for Feynman), diagonal-correctness quantises to multiples of $1/8 = 0.125$ (or $1/10 = 0.10$). The TOST margin δ=0.10 is at or below this quantum, meaning individual seeds cannot distinguish effects smaller than one form. We address this by using n=25 seeds: the *mean* across seeds is continuous, and the standard error ($\sim$0.02) is well below δ. All TOST conclusions rest on the cross-seed mean, not individual-seed resolution.

Table 4: Retrieval and classification metrics on the held-out synthetic test set. All three trained encoders achieve high retrieval mAP; the random encoder achieves chance classification accuracy (0.500) but nearly matches the trained encoders on equivalence-class retrieval mAP (0.893), indicating that variable-identity geometry already reflects algebraic proximity under any projection.

Encoder	Accuracy	mAP@10 (tree)	mAP@10 (equiv)
Tree-LSTM	0.925 ± 0.004	0.873 ± 0.050	0.906 ± 0.025
GIN	0.899 ± 0.013	0.862 ± 0.027	0.892 ± 0.010
Transformer	0.688 ± 0.170	0.865 ± 0.027	0.904 ± 0.007
BCE-only	0.932 ± 0.006	0.878 ± 0.021	0.880 ± 0.014
Random encoder	0.500 ± 0.009	0.887 ± 0.008	0.893 ± 0.012

Retrieval and Classification. All three trained encoders achieve classification accuracy above 0.88 and equivalence-class retrieval mAP@10 above 0.89. The BCE-only encoder achieves the highest classification accuracy (0.932) but lower equivalence-class mAP (0.880) than the Tree-LSTM, suggesting that the triplet objective specifically improves equivalence-class clustering at a marginal cost to classification accuracy.

The random encoder achieves 0.500 classification accuracy (chance) but mAP@10 (equiv) = 0.893 $\pm$ 0.012, nearly matching the trained encoders. The mAP@10 (tree) = 0.887 $\pm$ 0.008 is also high. This indicates that unsupervised tree-distance geometry already strongly reflects algebraic proximity. This is consistent with the variable-identity leakage finding: expressions that share variable indices are geometrically similar under any projection, including random ones. The fact that the corpus equivalence classes have size $\sim$3 (1999 of 2001 classes contain exactly 3 expressions) means that every positive pair is a single-rewrite paraphrase of its anchor with near-total token overlap: the easiest possible retrieval task.

The two mAP@10 variants measure different notions of retrieval success. The tree metric treats two copies of the same expression as positives regardless of paraphrase; the equivalence metric treats all paraphrases (including the expression itself) as positives. The fact that random encoders achieve high equivalence mAP but only chance classification accuracy shows that retrieval-level proximity and classification-level discrimination are separable objectives, and that the former is largely a function of variable-index overlap rather than learned algebraic structure.

B. External Validation: Operator-Token Leakage on a Symbolic-Mathematics Benchmark

We present this as *secondary validation* that the diagnostic ports to a benchmark and a trained model we did not build, not as a primary leakage result: as the caveat at the end of this section makes explicit, the family labels here are partly *defined* by operator content, so the audit demonstrates that the protocol transfers and that operator tokens are accessible in trained representations, rather than exposing a design flaw of the kind the main results document. With that scope fixed: the Feynman demonstration (Appendix D) concerns multi-variable physics equations, where variable identity is the dominant leak. To test the *second* leakage layer (operator-token leakage, i.e. S2 in §3.2) on a benchmark we did not construct, we turn to the integration dataset of Lample & Charton (2020) (the public `prim_ibp` split of their Deep Learning for Symbolic Mathematics benchmark). These are single-variable functions of x, so cross-expression variable-identity leakage *cannot* apply; any above-chance identification must come from operator tokens or tree shape.

We parse 320 real integrands from their test distribution into expression trees and label each by integration family: *polyrational* (only $+, -, \times, \div$ and integer powers), *radical* ($\sqrt{\cdot}$), *trig* ($\sin$ / $\cos$ and derived), and *explog* ($\exp$ / $\log$, with hyperbolic functions expressed via $\exp$), 80 per family. Powers are expanded via the paper's own elementary rewrites so that polynomials carry no transcendental tokens. We ask a random (untrained) Tree-LSTM, the same architecture used throughout the paper, to identify a held-out expression's family by nearest-family-centroid (25% held out per family, 50 random seeds). Chance is $1/4 = 0.25$. As an operator analogue of the within-range renaming control, the *operator-scramble* condition maps every binary operator to a single token and every unary operator to a single token, preserving tree shape but destroying operator identity.

Table 5: Random-encoder integration-family identification on Lample & Charton (2020) integrands ($n = 50$ seeds, chance $= 0.25$). Operator-scramble is the operator analogue of within-range renaming.

Condition	Family-identification accuracy
Natural (operators kept)	0.79 ± 0.07
Operator-scrambled control	0.58 ± 0.05
Chance	0.25
Per-family (natural):	
polyrational	0.80
radical	0.87
trig	0.82
explog	0.69

A random encoder identifies the integration family of published integrands at $3.2\times$ chance (0.79 vs 0.25) with no training. Scrambling operator identity drops accuracy to 0.58, a leakage index of $\ell = (0.79 - 0.58)/0.79 = 0.27$, confirming that roughly a quarter of the identification is carried by operator tokens directly; the residual above chance is tree-shape and unary-arity signal (also unlearned). The practical consequence mirrors the main paper: any learned equation-embedding model reporting family-level clustering on this benchmark must clear a 0.79 random-encoder skyline before its numbers can be credited to learned algebraic structure rather than operator-token statistics. This demonstrates the operator-token leakage layer on a published symbolic-mathematics corpus the authors did not construct.

Trained encoder audit (tag `r44_lample_charton_encoder`). We load the publicly available Lample & Charton (2020) Transformer encoder (80M parameters, 6 layers, d=1024, trained on mil-

lions of FWD+BWD+IBP integration problems) and run the same centroid-identification protocol on the same 320 integrands (n=25 seeds). Results:

- Natural accuracy: 0.876 ± 0.044 ($3.5\times$ chance; exceeds random encoder's 0.79).
- Operator-scrambled: 0.653 ± 0.055 ($p<0.0001$ vs natural, paired t=17.1).
- Operator-token leakage index: $\ell = (0.876 - 0.653)/0.876 = 0.254$.

Training at genuine structural scale (millions of diverse integrands) *amplifies* the family-identification signal, the trained encoder exceeds the random-encoder skyline (0.876 vs 0.793), yet operator-token reliance persists at a similar fractional rate (ℓ=0.254 vs 0.269). Operator scrambling still degrades identification significantly even in this massively trained model. The residual after scrambling (0.653, $2.6\times$ chance) reflects tree-shape and structural signal that survives operator removal and increases with training; the diagnostic cleanly separates the operator-token component from this structural component.

Interpretive caveat: label-definition confound. Unlike variable identity in the algebraic-equivalence benchmarks (which is genuinely notational and orthogonal to the intended algebraic computation), operator content *partially constitutes* the family labels used here. The four families (polyrational, radical, trig, explog) are defined by which operators appear; showing that operator scrambling degrades family identification is partly evidence that the model uses the feature that defines the ground-truth label, not purely evidence of a shortcut. This is qualitatively different from the variable-identity shortcut in the main result, where the shortcut exploits a design flaw rather than a semantically relevant feature. We retain this audit as evidence that (i) the diagnostic protocol ports to trained models at genuine structural scale, (ii) operator tokens are readily accessible in trained representations, and (iii) operator-token reliance is an inherent feature of operator-heterogeneous benchmark families. However, this result should be interpreted as *operator-content sensitivity* rather than as a direct analogue of the variable-identity leakage documented in the main paper.

C. Why Token-Identity Leakage Suffices: A Token-Bag Account

The leakage we document across three benchmarks admits **a simple *empirical* account of much, not all, of the random-encoder signal.** We state its limits first, because they are what the account is worth. **It is not a theorem, and a randomly initialised recursive network is *not* a random projection of the token multiset**: its output geometry carries tree topology, depth and nonlinear interactions too ("Dual-skyline divergence" below), and the account holds only where held-out forms have tree shapes similar to the training forms. Under substantial shape divergence it fails outright, and we give the counterexample below rather than filing it as a caveat. Within that domain, the account is: represent each expression by its *token bag*, the multiset of variable and operator tokens, ignoring tree structure. At random initialisation a tree encoder's output is dominated by an approximately linear aggregation of its input token embeddings, so expressions with similar token bags receive similar embeddings; and since random projection approximately preserves pairwise distances (a Johnson–Lindenstrauss property), nearest-centroid identification in the random embedding space tracks that in token-bag space. Where it applies, identification accuracy is then predicted *a priori* by token-bag separability, high exactly when a class's variable-set or operator-vocabulary is distinctive, near chance when it is not. This says nothing about learned structure (it holds for any random encoder) and it is what makes the random-encoder row a meaningful skyline.

We verify the account directly on the external benchmarks. A pure bag-of-tokens nearest-centroid classifier (no encoder, no training, operating on normalised token-count vectors) achieves 0.85 ± 0.05 family identification on the Lample & Charton (2020) integrands, matching and slightly exceeding the random encoder's 0.79; token-bag geometry accounts for the identification with room to spare. On the AI Feynman benchmark, all ten equations have distinct variable-bags, and the three with fully unique variable sets are the ones identified at 0.86 (versus 0.71 for the seven with shared variables, Table 8): exactly the ordering the token-bag account predicts.

The sharpest confirmation, however, is on the paper's *primary* evidence. Table 6 runs the same held-out-form protocol as the main suite in token-bag space, with no encoder at all. A *variable-aware* bag (distinct token per variable index) matches the trained Tree-LSTM's headline plain diagonal-correctness (Table 22) and *collapses to* 0.000 under within-range renaming: the trained-vs-random gap in the plain regime is explained by variable-token statistics under this protocol, with zero con-

tribution from tree structure (the bag has none). Note that Table 6 reports deterministic values: given the fixed 8-law registry with fixed expression forms, the classifier is a lookup with no randomness. With 8 laws, diagonal-correctness quantises to multiples of 0.125 (7 possible outcomes above chance), so the exact 0.875 match is less remarkable than it first appears. A *variable-blind* bag (all variables collapsed to one token) reproduces the random encoder's weaker plain score and is rename-invariant by construction. On the variable-leakage-proof overlap suite the variable-aware bag matches the random encoder's 0.40 and drops to chance (0.20) under renaming, while the variable-blind bag sits at chance throughout, confirming that the overlap suite's residual signal is operator/structure, not variable identity. The token-bag account thus explains the paper's own headline numbers, not only the external ones.

Table 6: Bag-of-tokens nearest-centroid skyline (no encoder, no training) under the held-out-form protocol. "Variable-aware" keeps a distinct token per variable index; "variable-blind" collapses all variables to one token. Compare to the trained Tree-LSTM and random encoder in Table 22.

Suite	Bag	Plain diag.	Within-range renamed
Main 8-law (chance $= 0.125$)	variable-aware	0.875	0.000
	variable-blind	0.375	0.375
Overlap 5-law (chance $= 0.20$)	variable-aware	0.400	0.200
	variable-blind	0.200	0.200

The practitioner's takeaway is an *a-priori* check: before training an encoder, compute token-bag separability across the intended classes; if it is high, a random encoder, indeed a structure-free token-count classifier, will already "identify" them, and any learned model must be measured against that skyline rather than against chance.

Validity domain. The JL + linear-aggregation account has a validity domain: it holds when the test form has similar tree shape to the training forms (as in E3, where held-out = library paraphrase within one rewrite step). Under substantial shape divergence (as in E3b, where held-out forms use out-of-library rewrites that change operator structure), recursive nonlinearity decouples the random encoder's output from the bag. The E3b counterexample is direct: on identical forms with identical centroid-construction protocol, the bag scores 0.900 and the random encoder 0.124. The discriminating factor is not centroid quality (the bag also uses 5-form centroids and is unimpaired), but tree-shape divergence: the bag accumulates tokens regardless of structure, whereas the random encoder's tree recursion maps structurally different trees to embeddings that are decorrelated from their token-bag representations. The practical lesson is to *report both skylines* (bag-of-tokens and random encoder), since they measure different failure modes and agree only in-distribution. When a held-out evaluation uses structurally divergent forms, the bag skyline remains meaningful but the random-encoder skyline may understate the token-bag ceiling.

Arity as a sub-bag cue. Variable-count (arity) is a token-bag property (it equals the number of distinct variable tokens after renaming). It is therefore a sub-type of token-bag leakage, not a separate phenomenon. However, it survives within-range renaming, which preserves variable *identity* but not arity, and so constitutes a confound in arity-heterogeneous benchmarks. An arity-controlled evaluation (restricting classification to same-arity groups) removes this confound. On E3b, the arity-controlled bag scores 0.375 renamed; the trained encoder scores 0.148 overall (and at or below within-arity chance), confirming that the trained encoder does not exceed the zero-parameter skyline even after arity deconfounding. The arity-aware bag protocol restricts nearest-centroid classification to equations sharing the same variable count: the 2-variable group (6 equations, chance 0.167) and the 1-variable group (2 equations, chance 0.500) are classified independently, eliminating arity as a discriminative cue while preserving the centroid-distance protocol.

D. Trained Encoder on AI Feynman: Full Results (E3)

Experiment E3 trains a Tree-LSTM on the AI Feynman corpus (10 equations, k=4 paraphrases each) and evaluates nearest-centroid identification on natural and within-range renamed forms across 25 seeds. The random-encoder-only results for the same corpus are given below, under *AI Feynman benchmark*; this section extends them to a trained encoder. Chance $= 1/10 = 0.10$.

Table 7: E3: trained vs. random encoder identification on 10 AI Feynman equations (25 seeds). Leakage index $\ell = 1 - \text{renamed}/\text{natural}$.

Encoder	Natural accuracy	Renamed accuracy	Leakage index
Bag-of-tokens (var-aware)	0.992 ± 0.027	—	—
Trained Tree-LSTM	0.972 ± 0.060	0.824 ± 0.086	0.15
Random encoder	0.724 ± 0.184	0.624 ± 0.145	0.14
Bag-of-tokens (var-blind)	0.636 ± 0.111	—	—
Chance	0.100	0.100	—

Replication, and why the headline uses the variable-only row. The skyline sweep (tag `r69`) re-runs *this same protocol* independently (same 10 equations, same k=4 paraphrases, same K=10, same n=25 seeds) and does not reproduce every row exactly: it gives full-token bag 0.932 ± 0.075 and random encoder 0.748 against the 0.992 ± 0.027 and 0.724 above, while the variable-blind bag agrees exactly (0.636). The bag rows move because the held-out paraphrase is redrawn, and the two full-token estimates are within one standard deviation of each other. But the difference straddles the trained encoder: $0.992 > 0.972$ in this run and $0.932 < 0.972$ in the replication, so the *direction* of a comparison resting on a 0.02 margin is not stable across runs. We therefore state the main-text claim (§4.1) against the *variable-only* bag, which `r69` measures at 1.000 with sd 0.000: with 100% unique variable sets on this suite (Table 23), a variable-token bag is a perfect lookup by construction, not a sampled estimate, and it exceeds the trained encoder under either run. This is the paper's own checklist applied to the paper's own headline number.

The variable-aware bag-of-tokens baseline achieves 0.992, exceeding even the trained encoder's natural accuracy (0.972). This means that *even on Feynman*, natural-condition identification does not clear the zero-parameter token-bag skyline. However, the trained encoder's *renamed* accuracy (0.824 vs random 0.624) under this lenient protocol appears to demonstrate structural learning that survives the removal of variable identity, but see "Protocol asymmetry" below and Appendix E for the E3b correction. The variable-blind bag (0.636) shows that operator-token leakage accounts for $\approx 6\times$ chance and closes most of the gap to the trained encoder's renamed accuracy (0.824).

Under this protocol (held-out = library paraphrase), both leakage indices are low (0.15 and 0.14). However, E3b (Appendix E) shows that this low-ℓ result is protocol-driven: when the held-out form is a genuinely out-of-library rewrite, ℓ rises to 0.71. The structural diversity of the Feynman corpus, equations ranging from Gaussians to Doppler factors to cyclotron frequency, produces a lenient E3 result that appears to show structural learning, but E3b and the arity-controlled analysis (Appendix E) show the apparent residual is explained by arity and operator-token cues rather than learned algebraic structure.

Protocol asymmetry (resolved by E3b). E3 evaluates nearest-centroid identification on one withheld paraphrase per equation (generated by the same paraphrase library used in training). This is analogous to the held-out-form protocol of §4.5 but not identical: in the main suite the held-out form is a hand-crafted algebraic alternative never produced by the paraphrase library, whereas E3's held-out form is a library-generated paraphrase seen neither during encoder training nor centroid computation. E3b (Appendix E) resolves this confound: under a protocol-matched evaluation using out-of-library held-out forms, ℓ rises from 0.15 to 0.71, confirming that the low-leakage E3 result was protocol-driven.

The finding has two implications. First, structural diversity is no protection: on a corpus ranging from Gaussians to Doppler factors, identification still reflects token shortcuts once evaluation uses genuinely out-of-distribution forms, and arity-controlled analysis (Appendix E) shows the apparent trained-vs-random residual is explained by arity and operator-token separability rather than learned algebraic structure, but this is *one* corpus, and **no prevalence follows**: the 23-corpus survey (§4.1) finds the S1 flaw confined to one family. Second, the token-bag skyline is *portable*, not universal, and licenses one direction only: any benchmark where bag-of-tokens exceeds reported accuracy has not demonstrated structural learning under natural evaluation, and renaming controls are required there, while clearing the bag certifies nothing.

External Validation: AI Feynman Benchmark. To test whether the variable-identity leakage mechanism generalises beyond our synthetic benchmark, we evaluate it on 10 equations from the

AI Feynman dataset (Udrescu & Tegmark, 2020): Gaussian (I.6.20), Coulomb force (I.12.5), gravitational PE (I.14.3), spring PE (I.14.4), capacitor voltage (I.25.13), wavenumber (I.29.4), cyclotron frequency (I.34.8), thermal energy (I.39.1), energy density (II.8.31), and Doppler factor (I.34.14, normalized). We map each equation's physical variables to unique token indices (16 symbols total) and encode each equation plus 4 paraphrases (commutativity, associativity) using a random (untrained) Tree-LSTM encoder. Identification is nearest-centroid over 50 random seeds; chance is $1/10 = 0.10$.

Table 8: Random-encoder identification on AI Feynman equations ($n = 50$ seeds). Natural = original variable assignment; renamed = variables permuted within range. Chance = 0.10.

Condition	Identification accuracy
Natural (original variables)	0.752 ± 0.171
Renamed (within-range permuted)	0.626 ± 0.163
Chance	0.10
By variable-set uniqueness:	
Equations with unique variable set (3/10)	0.86
Equations with shared variables (7/10)	0.71

A random encoder identifies AI Feynman equations at $7.5\times$ chance (0.752 vs 0.10) using variable identity alone. Equations with unique variable sets (Gaussian uses only x_0; gravitational PE uses $\{x_3, x_4, x_5\}$; spring PE uses $\{x_6, x_7\}$) are identified at 0.86, versus 0.71 for equations sharing variables with other laws. Within-range renaming reduces accuracy to 0.626 but does not eliminate it, because operator-token leakage persists (equations with distinctive operator vocabularies remain separable). This replicates the two-layer leakage structure found on our synthetic benchmark: variable-identity leakage dominates, operator-token leakage provides a secondary signal, and both operate without any training. We note that the residual 0.626 renamed accuracy conflates operator-token leakage with variable-count (arity) leakage: within-range renaming preserves the number of distinct variables per equation, so equations with unique arities (e.g., Gaussian uses one variable, gravitational PE uses three) remain distinguishable even after renaming. **What these 10 equations establish is bounded, and the unbounded version is the inference this paper exists to refuse.** They establish that variable-identity leakage is not an artefact of *our* benchmark design, since it reproduces on a benchmark we did not build. **They cannot establish that it is a general mechanism of neuro-symbolic tasks**: 10 equations are a constructed counterexample, and the prevalence evidence is the 23-corpus survey (§4.1), which finds S1 exposure *confined* to this one family.

E. Diversity Sweep Provenance (Table 21)

The eight per-equation values behind Table 21. Both rows are equation-level bootstraps over these eight numbers (n=8, n=25 seeds and three held-out forms per equation per row; tag `r63`). We print them because an interval computed from eight values should be auditable, and because their spread explains the interval widths: the in-library row is near ceiling and tightly clustered (SD 0.021), while the out-of-library row is genuinely dispersed (SD 0.162), which is why its interval is the wider of the two. The per-*form* ℓ' heterogeneity quoted elsewhere in this paper (-0.05 to 1.12) is a different quantity from per-equation plain accuracy and does not bound these widths.

	in-library held-out form			out-of-library held-out form		
Equation	plain	renamed	ℓ'	plain	renamed	ℓ'
gaussian	0.987	0.960	+0.031	0.760	0.680	+0.126
coulomb_force	0.947	0.813	+0.162	0.587	0.160	+0.924
grav_pe	0.973	0.893	+0.094	0.707	0.573	+0.229
spring_pe	0.973	0.853	+0.141	1.000	0.867	+0.152
capacitor_v	0.920	0.800	+0.151	0.987	0.800	+0.217
cyclotron	0.973	0.960	+0.016	0.907	0.747	+0.205
thermal_energy	0.973	0.907	+0.079	0.707	0.467	+0.413
energy_density	0.973	0.853	+0.141	1.000	1.000	+0.000
mean	0.965	0.880	+0.102	0.832	0.662	+0.283

Superseded measurements, kept for comparison. Two earlier constructions of the out-of-library endpoint are retained here so the correction in §4.5 is checkable. Tag `r33` swapped the out-of-library form into one equation's slot at a time, leaving in-distribution forms in the other seven, and reported $(0.545, 0.270)$ over eight equations; tag `r13` used one hand-crafted form per equation over all ten and reported $(0.504, 0.148)$. Both make the target equation compete against easier distractors than itself, which inflates the apparent protocol sensitivity. Table 21 uses neither.

Table 21 (the matched in-library/out-of-library pair) derives from one run family (n=25 seeds, identical hyperparameters; tags `r11_feynman_trained` for E3, `r13_feynman_matched` for E3b, `r16_feynman_middle` for E3m). Additional rows from earlier run families are listed below for historical reference. All use the Tree-LSTM encoder with identical hyperparameters (embed_dim=128, margin=0.2, 3 encoder epochs, 100 physics epochs with patience 10). Historical subset construction:

- **Overlap (shared vars):** The 5 overlap laws from `physics_laws_overlap.py`, all over variables x_0, x_1. Run family: `r8_overlap`, 5 seeds.
- **Full synthetic:** All 8 laws from `physics_laws.py`. Run family: `r9_heldout_ext`, 25 seeds.
- **4 unique-var laws:** Newton (x_0), Kepler (x_{10}), Pendulum (x_5, x_6), Ohm (x_8, x_9), chosen for disjoint variable sets (zero shared variables). Run family: `r12_diversity_cpu`, 5 seeds.
- **6 mixed laws:** Newton, Coulomb, Kepler, Hooke, Pendulum, Ohm, the 8-law suite minus kinetic energy and gravitational PE (chosen to vary the overlap structure). Run family: `r12_diversity_cpu`, 5 seeds.
- **AI Feynman (E3 protocol):** 10 equations from Udrescu & Tegmark (2020). Run family: `r11_feynman_trained`, 25 seeds. E3 protocol (Appendix D): held-out form is the last library paraphrase.
- **AI Feynman (protocol-matched, E3b):** Same 10 equations. Run family: `r13_feynman_matched`, 25 seeds. Held-out form is a hand-crafted SymPy-verified algebraic equivalent NOT producible by the paraphrase library. Training is identical (k=4 library paraphrases of the canonical form).

The subset selection for 4_unique and 6_mixed was pre-specified by the criterion "vary the degree of variable sharing" before seeing results, not chosen post-hoc to maximize or minimize ℓ. Bootstrap 95% CIs for ℓ use 10,000 resamples of the per-seed plain and in-range values.

E3b: Protocol-matched Feynman evaluation. The protocol-matched evaluation (E3b) addresses the confound disclosed in Appendix D: E3 holds out a library-generated paraphrase that shares the same tree structure up to one rewrite step, making identification "strictly easier" than the main suite's out-of-library held-out forms. E3b replaces this with 10 hand-crafted equivalent forms verified via `sympy.simplify(a - b) == 0`, each using algebraic rewrites absent from the paraphrase library:

Equation	Rewrite	Form
Gaussian	reciprocal-of-exp	$1/\exp(\theta^2/2)$
Coulomb force	half-of-double	$(qE_f + qE_f)/2$
Gravitational PE	scale-and-divide	$(2mgz)/2$
Spring PE	constant repositioning	$(k/2) \cdot x^2$
Capacitor	reciprocal multiplication	$q \cdot (1/C)$
Wavenumber	numerator doubling	$(\omega + \omega)/(2c)$
Cyclotron	fraction splitting	$(qv) \cdot (B/p)$
Thermal energy	constant decomposition	$(3pV)/2$
Energy density	factor rearrangement	$E_f \cdot (E_f/2)$
Doppler factor (I.34.14)	cross-multiplication	$c/(c - v)$

Note on the tenth equation. The canonical form $1/(1 - v/c)$ is a normalized version of AI Feynman I.34.14 ($\omega_0/(1 - v/c)$, Doppler factor) with $\omega_0 = 1$; it is used under that label throughout. Two earlier drafts mislabelled it: first as "Lorentz factor," then as "susceptibility (II.11.28)" (the published II.11.28 in Udrescu & Tegmark 2020 is the Clausius-Mossotti dielectric function $1 + n\alpha/(1 - n\alpha/3)$). The held-out form $c/(c - v)$ is algebraically equivalent:

$c/(c-v) \cdot (1/c)/(1/c) = 1/(1-v/c)$. The canonical expression used in training is correct; experimental results are unaffected by the label.

Results (n=25): trained plain 0.504 ± 0.087, trained renamed 0.148 ± 0.085, random plain 0.124 ± 0.065, random renamed 0.044 ± 0.050. Trained renamed is significantly above chance (p=0.011, one-sample t-test vs 0.10) and significantly above random renamed ($p<0.001$, paired t), but at $1.5\times$ chance rather than E3's $8.2\times$. The leakage index ℓ=0.71 $[0.64, 0.77]$ confirms that most of E3's apparent structural advantage was protocol-driven.

However, the trained-vs-random contrast (0.148 vs 0.044) does not constitute a structural residual by the paper's own skyline standard. Arity-controlled analysis reveals that the 0.148 overall is almost entirely explained by arity and operator-token cues: within the 2-variable group (6 equations, chance 0.167), trained renamed is 0.067 ($0.40\times$ chance, below chance); within the 1-variable group (2 equations, chance 0.500), trained renamed is 0.440 ($0.88\times$ chance). The arity-aware bag classifier scores 0.375 under the same within-range renaming, well above the encoder's 0.148. In other words, a zero-parameter arity-aware bag outperforms the trained encoder on the same evaluation. The trained encoder is less pathological than the random encoder (which scores 0.044 due to systematic mismatch; see §Random-renamed below chance), but the residual reflects arity and operator-token separability, not learned algebraic structure.

Bag-of-tokens skyline on E3b. The variable-aware bag-of-tokens classifier (same protocol: centroids from canonical + k=4 paraphrases, nearest centroid) achieves 0.900 on E3b plain, far above the random encoder's 0.124, confirming that variable-set distinctiveness is preserved in the held-out forms and suffices for identification. Under within-range renaming, the bag drops to 0.400: not chance (0.10) because equations differ in variable *count* (1-var, 2-var, 3-var, 4-var), creating arity-distinguishable bags even after index remapping.

Arity-controlled bag. Restricting classification to same-arity groups (excluding trivial singleton groups of arity 3 and 4): variable-aware bag achieves 0.375 renamed (0.375 plain); variable-blind bag achieves 0.375 renamed (0.375 plain). The arity-controlled variable-blind score of 0.375 reflects operator-token separability that survives both renaming and arity deconfounding. As shown above, the trained encoder scores only 0.148 overall (and below within-arity chance), confirming that it falls short of the zero-parameter arity-blind operator-bag skyline.

Why the random encoder collapses on E3b. The bag's 0.900 on E3b plain contrasts with the random encoder's 0.124 on identical forms with identical centroid protocol. If the JL + linear-aggregation account from Appendix C held, the random encoder would track the bag and score near 0.90. That it does not reveals a validity-domain condition: the token-bag account holds when tree shapes are similar (E3's single-rewrite paraphrases) but breaks under substantial shape divergence. E3b's out-of-library rewrites produce different operator *structure*, not just different operator bags; the tree encoder's recursive nonlinearity decouples from the linear bag aggregation when tree shapes differ markedly between training and test forms. The bag builds centroids from the same five forms as the random encoder but is unaffected by shape divergence (it aggregates tokens regardless of structure), so centroid quality is identical for both: the collapse is a tree-structure effect, not a centroid-quality effect. See Appendix C for the full account.

Random-renamed below chance. The random encoder under E3b renaming scores 0.044 ± 0.050, significantly below chance (0.10; one-sample $t = -5.6$, $p<0.001$). This is not sampling noise but a systematic effect of operator-content mismatch. After renaming, equations group by variable-count: 2 use one variable, 6 use two, 1 uses three, 1 uses four. Within each group, all equations share identical variable sets, so classification depends on operator tokens. Because E3b's held-out forms use out-of-library rewrites, their operator distributions differ systematically from their own centroid's training forms. Under random projection, this mismatch pushes the held-out embedding closer to *other* equations' centroids that share more operator tokens with the test form, producing systematic misassignment below chance. The trained encoder avoids this pathology (0.148, p=0.011 above chance) because training induces structure-sensitive embeddings that partially resist the operator-content mismatch. A below-chance control deserves more than an explanation, so we state its consequence explicitly: an accuracy reliably below $1/K$ means the skyline is *anti*-correlated with the truth, not merely uninformative; under a permutation of the class labels the expected accuracy is exactly $1/K$ by construction, so no label-permutation null can produce 0.044. We therefore do not

treat the random-renamed value as a floor anywhere in the paper: comparisons under E3b renaming are made against chance and against the variable-blind bag, both of which are well-defined, and the random-encoder skyline is reported for the plain condition only.

SymPy-Verified External Forms. Five external forms covering five of the 8 laws (Newton gravity, Hooke, kinetic energy, Ohm, pendulum) were generated by algebraic transformations NOT in the paraphrase library and verified equivalent via SymPy. Each form was assigned to the nearest of the 8 law clusters under Euclidean distance in the encoder's latent space. Per-item chance under uniform assignment is $1/8 = 0.125$; with only 5 items, identification accuracies quantise to multiples of 0.2.

Table 9: Identification accuracy for SymPy-verified external forms. The random encoder scores 0.36, well above chance, because Hooke and Ohm have unique variable-index sets, exposing the same variable-identity leakage the renaming controls demonstrate in the main suite.

Encoder	Identification accuracy
Tree-LSTM	0.68 ± 0.11
GIN	0.60 ± 0.20
Random encoder	0.36 ± 0.09

The random encoder scores 0.36, well above chance (0.125), because Hooke ($k \cdot x$) and Ohm ($I \cdot R$) have unique variable-index sets that any encoder, including an untrained one, deterministically identifies. This is the same variable-identity leakage the renaming controls expose in the main suite: even an untrained encoder "recognises" laws whose variables are unique within the training corpus. Trained encoders score higher here only where variable sets happen to differ across laws.

Token-Similarity Alignment Check.

Purpose. Under E3b, the trained encoder identifies 0.50 of held-out forms correctly, above chance (0.10) but far below the in-library rate. Is this residual driven by token co-occurrence (the encoder predicts whichever equation shares the most operator/variable tokens with the query) or by genuine algebraic-structure sensitivity? We test this with a token-similarity alignment check.

Method. For each of the 10 E3b held-out forms, we compute the bag-of-tokens cosine similarity to each equation's training centroid. A "token-similarity oracle" assigns each held-out form to the most bag-similar centroid (accuracy: 0.90, i.e. 9/10 forms are most bag-similar to their own equation). We then measure whether the trained encoder's actual predictions *agree* with the oracle's predictions (not whether both are correct) across 25 seeds.

Null model. Under independence, the expected agreement between an encoder with accuracy a and an oracle with accuracy b, assuming errors are distributed uniformly over $K-1$ alternatives, has exactly two contributing events: both correct (probability $a \cdot b$), or both wrong and landing on the same wrong class (probability $(1-a)(1-b)/(K-1)$). With a=0.504 (empirical encoder accuracy), b=0.90, K=10:

$$P_{\text{agree}} = a \cdot b + \frac{(1-a)(1-b)}{K-1} = (0.504)(0.90) + \frac{(0.496)(0.10)}{9} = 0.4536 + 0.0055 = 0.459.$$

We test whether observed agreement exceeds this baseline.

Results (n=25 seeds, tag `r40_family_scramble`).

- Token-similarity oracle accuracy: 0.90 (9/10 equations distinguishable by bag alone).
- Trained encoder E3b accuracy: 0.504 ± 0.09 (ground-truth identification).
- Observed encoder–oracle agreement: 0.604 ± 0.09.
- Expected agreement under independence: 0.459.
- One-sample t-test: t=8.15, p<0.0001 (n=25).

The encoder's predictions align with the token-similarity oracle significantly more than independence predicts. Token co-occurrence patterns (which operators/variables appear) contribute materially to E3b residual identification; the alignment check does not rule out additional algebraic

Table 10: **What running the audit changes, in all three modalities. Eight rows are already-published numbers, and four of those are systems and leaderboards built by other people**; the last two are claims of *ours* that only this revision's audit settled. The audit revises in *both* directions.

Claim as audited	Level	Verdict after the audit
symbolic mathematics: as published		
Feynman held-out 0.972	S1	**broken**: variable bag 1.000, 84% unique var. sets
score_5 leaderboard, 14 corpora	S1, S2	**upheld**: no non-learned floor reaches EqNet/SemEmb (App. V)
StructEmb ablation, 14 corpora	S1, S2	**narrowed**: an *untrained* encoder reaches it on 5
Lample–Charton 80M integration	S2	**S2 only**: operator/arity bag 0.941 vs. 0.872
Our poly8 0.894 and composition	S3, C	**restated**: passes $\mathcal{F}_3$ at $K \geq 200$, over what is tested; margin $1.7\times$ falls to $1.3\times$ under a search of ours (App. BB)
Our depth curve and chance level	—	**narrowed** and **corrected**, both *against* us
Our `r91` alternate-order $+0.250$	—	**withdrawn**: an aggregation artifact; ≤ 0.035 within a library
Our NeSymReS $\ell' = 0.44$	S1	**withdrawn**: a null under a global permutation
natural language: their grammar, their split, our encoder and our audit		
Our SCAN `add_prim_jump` 0.987	S3	**narrowed**: it *passes*, but $\sup \mathcal{F}_3 = 0.0005$: *their* held-out design puts the bar on the floor, so we quote no ratio (App. AY)
code: our corpus, built from public Python		
Our Type-2 clone S2 0.987	S2	**settled below S3**: *every* $\mathcal{F}_3$ member attains 0.990, the maximum this corpus admits

sensitivity but establishes that token-bag similarity is a significant component. Note: this analysis is scoped to E3b; E3m instability (4 equations, high per-equation variance) prevents analogous claims for the intermediate regime.

Provenance. Code: `run_family_scramble.py`. Data: `logs/r40_family_scramble.json`.

F. What the Audit Changed: the Ledger, and the Defect Log

The top-level Reproducibility Statement follows the conclusion; this appendix records the details behind it. All reported numbers are traceable to provenance tags (Appendix K). The diagnostic protocol was applied first to the authors' own results, where it identified and corrected multiple initial over-claims before submission; the protocol's sensitivity is calibrated by this internal use. Code and run logs are included in the supplementary material.

Read this as verification, not as a bug count. Every empirical claim in this paper was re-audited under corrected procedures, and the outcome column is what matters: **of the seven incidents, two changed a reported *number* without changing the finding it supported, two retracted or excluded a form, two, the tree-edit defect and the `r91` aggregation defect, corrected something *in this paper's favour* and are reported anyway, and the newest withdraws a portability claim we had made for ourselves.** None reversed a conclusion the paper draws: the `r91` correction *removed* a caveat we had published, and the seventh removes a claim. Table 11 is the whole revision history at a glance: seven incidents, each caught by the paper's own screens rather than by a reader. **The developmental chronology behind these rows (how each was found, what it cost, and the two later corrections that are not incidents but incomplete fixes) is `REVISION_HISTORY.md` in the supplement**, moved there so this appendix carries the runs and their outcomes. Every corrected *number* is in the paper, in the table it belongs to.

Revision note on §N's circularity check. Row 1 of Table 11: the retracted draft number was renamed $= 0.212 \pm 0.098$, p=0.008, and no run tag ever produced it. The first genuine evaluation of that cell replaced it, and every number in this paper has named its log since (Appendix K).

Multi-form E3b: schema-based OOL forms (tag `r22_schema_multiform_e3b`). To establish that the E3b leakage estimate $\ell' = 0.88$ is form-agnostic, we implemented nine mechanical rewrite schemas (half-of-double: $a \to (a{+}a)/2$; scale-and-divide: $a \to (2a)/2$; triple-of-third: $a \to (3a)/3$; add-subtract: $a \to (a{+}a){-}a$; reciprocal-multiplication: $a/b \to a{\cdot}(1/b)$; numerator-doubling: $a/b \to (a{+}a)/(2b)$; fraction-splitting: $(ab)/c \to a{\cdot}(b/c)$; fraction-split-

rev: $(ab)/c \to b{\cdot}(a/c)$; constant-repositioning: $(ka)/b \to (k/b){\cdot}a$) as an automated OOL generator (`schema_ool_generator.py`). All schemas are provably absent from the paraphrase library $\mathcal{R}$. The generator applies each schema at every eligible sub-term of the canonical expression tree, syntactically deduplicates against the canonical and hand-crafted held-out, then verifies algebraic equivalence via `sympy.simplify(canonical - candidate) == 0` with a 3-second timeout. Result: all 10 Feynman equations yield exactly 5 verified OOL forms each (50 forms total). Each form was evaluated under the E3b protocol (baseline encoder, 25 seeds), replacing that equation's hand-crafted held-out form while the other 9 equations keep their original forms. *Results (tag `r22_schema_multiform_e3b`, 50 form/equation pairs, 25 seeds each):* ℓ' values across all 50 pairs: mean $= 0.778$, median $= 0.750$, std $= 0.088$, range $[0.647, 0.926]$. The hand-crafted baseline is ℓ'=0.88 $[0.80, 0.94]$. The schema-generated forms span a distribution centred around the hand-crafted estimate: 44 of 50 forms yield $\ell' > 0.65$, and 22 of 50 yield $\ell' > 0.83$. The variation within an equation across forms (e.g. gaussian: 0.705–0.873) reflects that some schema applications produce forms structurally closer to the training paraphrases than others, reducing the distributional shift. No form yields $\ell' < 0.65$, confirming that high leakage is robust across mechanically-generated OOL forms. The distribution of ℓ' confirms that the OOL leakage finding is not artefactual to the hand-crafted forms: automated schema-based forms spanning all 10 equations consistently reproduce high leakage ($\ell' \gg 0$).

Natural OOL rewrites (tag `r26_natural_ool_rewrites_v2`). A reviewer raised the question of whether OOL forms constructed by practitioners, using standard algebraic manipulation rather than mechanical schemas, would also show high leakage. We applied SymPy algebraic strategies (factor, partial fractions, explicit reciprocal) and node-level structural rewrites (product reassociation, factor-to-denominator) to the 10 Feynman canonical expressions, excluding any form matching the canonical, training paraphrases, hand-crafted held-out, or mechanical schemas.

Deduplication audit. Row 3 of Table 11: a preliminary run (tag `r26_natural_ool_rewrites`, 13 forms) admitted doppler_factor's SymPy `factor` form $x_{11}/(x_{11} - x_{12})$, structurally identical to the hand-crafted held-out $c/(c - v)$, because the deduplication logic checked string distinctness and not SymPy structural equality. A structural-identity check now runs wherever the held-out's SymPy form differs from the canonical (doppler_factor alone; elsewhere SymPy normalises the two together, so the check would exclude every valid rewrite), leaving 12 forms.

Result: 12 verified natural OOL forms across 8 of 10 equations, evaluated under the E3b protocol (tag `r26_natural_ool_rewrites_v2`, n=25 seeds each, derived from tag `r26_natural_ool_rewrites` by excluding the doppler/factor duplicate). *Results:* ℓ' mean $= 0.800$, median $= 0.813$, std $= 0.065$, range $[0.705, 0.885]$. All 12 surviving forms were verified non-equivalent to their equation's hand-crafted held-out via the structural-identity check, and algebraically equivalent to the canonical via `sympy.simplify(form - canonical) == 0`. The natural-OOL distribution matches the mechanical-schema distribution (mean 0.778) and lies within the hand-crafted range $[0.80, 0.94]$, confirming that high leakage is not an artefact of how OOL forms are constructed.

Self-validation note (r20 incident). Row 2 of Table 11; the retracted run is tag `r20_feynman_multiform_e3b`, and the equivalence check that caught it now runs before every held-out set is used.

Triplet-loss reconciliation (tags r17, r18). Both the baseline Feynman encoder (r17) and the alpha-augmented Feynman encoder (r18) exhibit triplet loss $= 0.0000$ by epoch 6 (identical training configuration). This does *not* imply encoder collapse. A prior draft cited zero triplet loss as evidence of representation collapse; that inference was incorrect: r18's E3 diagnostic shows E3-plain $= 0.95$ under zero triplet loss, confirming the encoder learned correctly within the library. Zero triplet loss on a small corpus (500 Feynman training samples) means all training triplets are satisfied: the encoder has memorised all in-distribution relationships. The failure is *transfer*: this memorisation does not extend to structurally novel out-of-library forms (E3b-plain $= 0.12$).

Synthetic OOL transfer test (tag `r19_alpha_ool_synthetic`). To evaluate transfer on a *functioning* (non-collapsed) alpha-augmented encoder, we trained both the alpha-augmented and baseline encoders on 8 synthetic laws (same recipe as Table 17, 2000 expressions, 3 epochs)

and evaluated both on 5 hand-crafted OOL synthetic forms from the synthetic benchmark (Appendix E; all 5 are algebraically equivalent to their canonical forms and verifiable by inspection; 2 of the 5 additionally pass automated `sympy.simplify(a-b)==0` verification, namely newton_gravity and hooke; the other 3 involve representation differences (e.g. $mv^2/2$ as a tree `div(mul(mul(m,v),v),2)` versus $\frac{1}{2}mv^2$) that SymPy's `simplify` does not resolve within the timeout, a known limitation of the automated check). Results (n=5 seeds; chance = 0.125): alpha OOL plain = 0.52 ± 0.10, renamed = 0.28 ± 0.10, $\ell' = 0.61$; baseline OOL plain = 0.68 ± 0.16, renamed = 0.40 ± 0.13, $\ell' = 0.51$. The alpha-augmented encoder is not collapsed ($4.2\times$ chance plain), confirming it functions correctly. However, alpha augmentation does not reduce OOL leakage: $\ell'_\alpha = 0.61$ vs $\ell'_{\text{base}} = 0.51$ (directional; n=5 seeds). This is consistent with the Feynman TRANSFER_FAILURE result (r18): variable-invariance learned within the training distribution does not transfer to OOL forms, regardless of whether the encoder collapses or not.

G. Tree-Edit-Distance Analysis

Algorithm. We compute the tree-edit-distance (TED) between expression trees using a recursive dynamic-programming algorithm on ordered labelled trees with unit costs (insert = delete = rename = 1). Each node in the expression tree carries a label derived from its kind and content: variables are labelled `var_`i (where i is the variable index), constants are labelled `const_`v (rounded to 4 decimal places), and operators are labelled `unary_`f or `binary_`$\circ$. Children are ordered (left before right for binary nodes). **This is a *constrained* (alignment) tree edit distance, not Zhang–Shasha** (Zhang & Shasha, 1989), and an earlier draft described it as a simplified variant of one: roots are always matched to roots and children aligned as sequences, so it never deletes an internal node and promotes its children. It is therefore an *upper bound* on the unconstrained distance, measured over 1500 random `poly8` pairs it is $\geq$ Zhang–Shasha on every pair and equal on 12, exceeding it by 3.05 on average. Both are legitimate distances and nothing in this appendix changes, but it is a *different quantity* from the order-sensitive Zhang–Shasha used for the twin in Appendix AB, and the two must not be read across.

Protocol. For each held-out form, we compute the minimum TED to its nearest training form (the canonical expression plus k=4 paraphrases generated with seed 0). $\text{TED}_{\min}$ is deterministic (Zhang–Shasha on fixed tree pairs; no random seed is involved). Lower TED indicates a held-out form that is structurally closer to the training distribution.

Sample. 74 data points from four protocols:

- **E3 library forms** (n=10): one library-generated paraphrase per equation held out (seed 999); aggregate ℓ'=0.17 over all ten equations (tag `r21`; note this is the 10-equation figure, not the TED-cleaned eight of Table 21).
- **E3m in-library forms** (n=4): exp/log/power rewrites for the 4 eligible equations; aggregate ℓ'=0.41, **an unstable probe rather than an intermediate regime, and no claim in the body rests on it**: the four per-equation values are 0.053, clipped >1, −0.686 and undefined (*E3m instability*, below). **The robust result here is the two-point E3/E3b contrast.**
- **E3b hand-crafted forms** (n=10): expert-written held-out forms; aggregate ℓ'=0.88 over all ten equations (tag `r13`; again the 10-equation figure).
- **Schema-generated OOL forms** (n=50): 5 forms per equation from `schema_ool_generator.py`; per-form ℓ' from tag `r22_schema_multiform_e3b` (25 seeds each).

Results (tag `r23_tree_edit_distance`).

Interpretation. The key result is that *the training-coverage boundary, not continuous structural distance, governs leakage*. Three pieces of evidence converge on this conclusion:

(1) TED ranges overlap. E3 library forms span TED $[0, 5]$ and E3b hand-crafted OOL forms span TED $[3, 8]$. The ranges overlap substantially: thermal_energy's E3b form has TED= 3 (lower than gaussian's E3 form at TED= 5), yet it shows ℓ'=0.88 vs ℓ'=0.17. The clearest example: thermal_energy E3b (TED= 3, ℓ'=0.88) vs gaussian E3 (TED= 5, ℓ'=0.17): a form with lower TED shows drastically higher leakage, purely because it lies outside the paraphrase library.

(2) TED=0 E3 forms. Two E3 library held-out forms (wavenumber and doppler_factor) have TED$=0$ to a training form, meaning the encoder saw a structurally identical expression during training. Identification is trivially solvable for these forms. This mechanically explains E3's low $\ell'=0.17$: the protocol admitted training duplicates into the held-out set. This is not a bug in the TED analysis but a revelation about E3's leniency.

(3) In-library but high-TED E3m forms. The 4 E3m exp/log/power rewrites are *in the paraphrase library* yet have high TED (mean 6.3, range $[3, 8]$), overlapping the E3b OOL range. They show $\ell'=0.41$ (intermediate). This directly falsifies the prediction that "high TED implies high leakage." A form at TED$=7$ inside the paraphrase library ($\ell'=0.41$) shows far less leakage than a form at TED$=3$ outside the library ($\ell'=0.88$).

Together, these three observations establish that the protocol boundary operationalises the training-coverage boundary, not structural distance. Within the OOL cluster ($n=50$ schema forms with individual ℓ' estimates), TED does not predict leakage variation: $r=-0.06$, $\rho=-0.13$ (both non-significant at $\alpha=0.05$). The Mann-Whitney U test confirms that in-library and OOL TED distributions are statistically separated ($p<0.001$), but this reflects an indirect correlation: OOL forms tend to have higher TED because they are designed to be structurally different, not because TED drives leakage. What does track it is whether the encoder's training distribution included the held-out tree structure, the variable the leave-one-schema-out intervention manipulates directly (Appendix W), and the only place in this paper where that relation is established by intervention rather than by correlation.

Full data table. Tables 14 and 15 together report all 74 data points (70 original + 4 E3m).

Per-equation ℓ'. Table 16 reports per-equation ℓ' for E3 (from tag `r21_feynman_trained_perequation`; $n=25$ seeds, 10-way classification) and E3m (from tag `r16_feynman_middle`; $n=25$ seeds, 4-way classification over the 4 eligible equations). Per-equation values show wide variation: wavenumber has $\ell'\approx1.12$ because its held-out form has TED$=0$ to a training form (identification is trivially solvable; renaming destroys the only cue). Negative ℓ' (e.g. energy_density E3m) indicates renamed accuracy exceeds plain, noise-dominated at the single-equation level. Doppler has E3m plain $= 0.000$ (the encoder completely fails to identify this form even without renaming), rendering ℓ' undefined.

E3m instability: The four E3m per-equation values span an extreme range: gaussian 0.053, spring_pe > 1 (clipped), energy_density -0.686, doppler undefined. The aggregate $\ell'=0.41$ is thus a mean over one near-zero, one maximal, one negative, and one degenerate equation: it should not be interpreted as a coherent "intermediate regime" but rather as an unstable probe whose mean happens to fall between E3 (0.17) and E3b (0.88). The robust claim is the *two-point* E3/E3b contrast (0.17 vs 0.88, both computed over 10 equations with stable per-equation behavior); the E3m point is consistent with the ordering E3 $<$ E3m $<$ E3b but carries high per-equation instability that prevents claims about the functional form. Per-equation E3b values are not available because the 10-equation centroid protocol evaluates all equations jointly (each equation's identification depends on all 10 centroids).

Provenance. Code: `run_tree_edit_distance.py` in the supplementary material. Data: `logs/r23_tree_edit_distance.json`. The r22 schema forms have individual per-form ℓ' because each is evaluated in a one-form-swapped design (only the target equation's held-out is replaced; the other 9 retain their hand-crafted forms). The within-OOL TED range for hand-crafted E3b forms is narrow (TED $\in [3, 8]$); the broader schema-generated set ($n=50$, TED $\in [5, 18]$) provides more range, and the null correlation ($\rho=-0.13$, n.s.) is not an artefact of restricted range.

H. Hardened Training Recipe

Motivation. The hardened recipe addresses the objection that our baseline Feynman training (10 epochs, single-rewrite paraphrases) is too weak to learn invariant features, and that a more aggressive augmentation strategy might overcome the variable-identity shortcut.

Recipe parameters.

- **Paraphrase generation:** k=8 chained paraphrases per equation via `paraphrase_chain(tree, k=8, depth=4, rewrite_list=_REWRITES_WITH_ALPHA_FULL)`. At each of the 4 chain steps, a random subtree node is selected and a random rewrite is applied from the 13-element list (12 structural rewrites: commutativity, distributivity, identity insertion/removal, exp/log, power rewrites; plus 1 full random-permutation alpha rename over all variable indices in the subtree). Alpha renaming is thus applied with probability $\approx 1/13$ per step, and to a random subtree rather than the full tree. Most positives receive 0–1 alpha-rename operations; the dominant augmentation is structural.
- **Training:** 20 encoder epochs (vs. 10 baseline), n=25 independent seeds.
- **Corpus:** 10 Feynman equations $\times$ 5 base forms $\times$ 10 variable assignments $=$ 500 base samples; chained paraphrases expand to $500 \times 8 = 4{,}000$ augmented positives.

Results.

- **E3 (in-library):** plain $= 0.992 \pm 0.040$, renamed $= 0.660 \pm 0.091$, ℓ'=0.37 $[0.33, 0.41]$.
- **E3b (out-of-library):** plain $= 0.480 \pm 0.071$, renamed $= 0.104 \pm 0.084$, ℓ'=0.99 $[0.91, 1.08]$, the upper limit exceeds 1 because renamed accuracy falls below chance in some resamples.

Near-perfect in-library identification shows the encoder learns from the augmented data; the simultaneous ℓ'=0.99 out-of-library shows that **the learned invariance does not transfer to the tested out-of-library forms under this protocol**, a statement about these forms and this protocol, not a demonstration that the representation is variable-identity dependent in general.

Provenance. Code: `run_hardened_recipe.py`. Data: `logs/r31_hardened_recipe.json`.

I. External Portability: NeSymReS, and a Withdrawal

Status of this result, and a correction we make against ourselves. This is secondary validation that the diagnostic machinery ports to a published model we did not build; it is *not* evidence for the leakage claim, and under the correct control it is a **null**. Column permutation is not the analogue of symbolic variable renaming (see the caveat below), so nothing in this subsection is counted toward any S1 or S2 finding in the main text; it establishes portability of the method, and that is all. **Earlier versions of this appendix reported a significant drop under column permutation (ℓ'=0.44, tag `r28`); that arm is *withdrawn***, for the reason given under *Results* below.

Purpose. To demonstrate that the diagnostic protocol ports beyond our training pipeline, we apply it to NeSymReS (Biggio et al., 2021), a published Set-Transformer encoder for symbolic regression (ICML 2021).

Protocol. NeSymReS maps numerical point-sets (X, y) to encoder embeddings via induced set-attention blocks; it shares no code, architecture, or training data with our pipeline. We extract encoder embeddings from the published 10M-parameter checkpoint for 10 Feynman equations (500 points each, 25 random seeds) and run the centroid-matching identification protocol. The analogue of within-range variable renaming for numerical inputs is *column permutation*: cyclically shifting which input column corresponds to which physical variable. **Two such controls are possible and they are not equivalent.** Permuting *only* the held-out samples changes the query's column convention while leaving every class centroid in the original one, so any drop it produces conflates encoder sensitivity with a broken centroid–query correspondence. Permuting *globally*, one fixed permutation applied to every centroid sample and every held-out sample, leaves the corpus *isomorphic* to the original, so a function-level encoder *must* be invariant and any drop is genuine notational sensitivity. **We report the global control.**

Results (global control, tag `r62`): a null. Plain accuracy $= 0.624$ $[0.564, 0.684]$, globally column-permuted accuracy $= 0.648$ $[0.612, 0.684]$, paired $\Delta = +0.024$ $[-0.036, 0.084]$ (10 equations, chance $= 0.10$, 25 seeds, one permutation drawn per seed). **The interval contains zero and**

the point estimate is the wrong sign for a shortcut: under a control that preserves the isomorphism, this published encoder shows *no* column-convention sensitivity.

Withdrawn: the held-out-only arm (tag `r28`). Earlier versions of this appendix reported column-permuted accuracy $= 0.392$ and $\ell' = 0.44$ from a control that permuted the held-out sets alone. **We withdraw that arm and every statement resting on it.** What it measured is the centroid–query mismatch described under *Protocol*, not a property of NeSymReS. `r28` is retained in the run index (Appendix K) marked as superseded; its *plain* accuracy of 0.624 reproduces exactly under `r62`, which is what localises the fault to the control arm rather than to the measurement.

Interpretive caveat, which now cuts the other way. Unlike symbolic variable tokens (which are purely notational and should be interchangeable for algebraically equivalent expressions), column position in numerical inputs carries legitimate semantic information: it binds physical variables to input dimensions. A model that were invariant to a permutation of *some* inputs and not others would conflate physical quantities and be broken. Under the *global* permutation, however, the corpus *is* isomorphic to the original, every sample relabelled the same way, so invariance is the correct behaviour and **the null is the expected result rather than a disappointing one**. What this subsection establishes is accordingly narrower than the withdrawn arm claimed, and is all we now claim from it: **the diagnostic protocol runs unmodified on a published numerical-input encoder and returns a well-behaved answer there**. It supplies no evidence of leakage in that model, and no result in the main text rests on it.

Provenance. Code: `run_r62_nesymres_global_permutation.py` (global control; supersedes `run_external_audit_nesymres.py`). Data: `logs/r62_nesymres_global_permutation.json`; the withdrawn arm is `logs/r28_external_audit_nesymres.json`.

J. Evaluation-Head Robustness

Purpose. Given that Table 19 shows the evaluation head can flip conclusions about training objectives, we verify that the headline ℓ' metric is stable across identification heads.

Method. We train 25 encoders (same protocol as the main E3b experiment) and evaluate each under three identification heads: nearest-centroid, kNN with k=3, and kNN with k=5. For each head, we compute ℓ' with bootstrap 95% CIs.

Results.

Head	Plain	Renamed	ℓ'	95% CI
Centroid	0.504	0.148	0.88	[0.80, 0.96]
kNN (k=3)	0.508	0.168	0.83	[0.73, 0.94]
kNN (k=5)	0.500	0.164	0.84	[0.75, 0.93]

All three heads produce ℓ' within ± 0.05 of each other (CIs overlap substantially). The leakage index is not an artefact of the centroid evaluation choice. Note that kNN slightly lowers ℓ' because it is more sensitive to local embedding geometry, but the qualitative conclusion (high leakage, $\ell' > 0.8$) is unchanged.

Provenance. Code: `run_head_robustness.py`. Data: `logs/r41_head_robustness.json`.

K. Run-Tag Provenance

All reported results derive from tagged experiment runs. Each tag names the script, configuration, and random-seed family used; logs and checkpoints are archived under `logs/<tag>.json`. The table below covers the AI Feynman and corpus-survey families. Every run from `r70` onwards, the EQNET `poly8`/`boolean8` experiments and the later ports to other domains, is tagged instead

in the appendix subsection that reports it, because their designs are inseparable from the guards reported alongside them; `REPRODUCE.md` in the supplement lists every tag of both groups in one place, with its command and runtime.

Tag	Result	Section
r8_overlap	Overlap-law baseline	App. E
r9_heldout_ext	Extended held-out forms	—
r11_feynman_trained	Table 22 (E3, n=25)	§4
r12_diversity_cpu	Calibration table (historical)	—
r13_feynman_matched	E3b matched evaluation (n=25)	App. E
r16_feynman_middle	E3m middle protocol (n=25)	App. D
r11/r13/r16_feynman	10-equation E3/E3b/E3m measurements	App. E
r17_feynman_alpha_matched	Alpha-augmented Feynman transfer	App. F
r19_alpha_ool_synthetic	Alpha OOL synthetic transfer	App. F
r20_feynman_multiform_e3b	Multi-form E3b (n=25)	App. F
r21_feynman_trained_perequation	Per-equation E3 (n=25)	§4.5
r22_schema_multiform_e3b	Schema-generated OOL forms	App. F
r23_tree_edit_distance	Tree-edit-distance analysis	App. G
r26_natural_ool_rewrites	Practitioner-style OOL rewrites	App. F
r30_crossover_manipulation	Crossover DID (n=25)	§4.5
r31_hardened_recipe	Hardened training recipe	§H
r34_scale_20k_powered	20K augmentation-volume point (n=25)	§M
r35_triplet_harm_factorial	Table 19 (triplet ablation, n=25)	§4
r36_positive_control	Positive control (ℓ'=0.08, n=25)	§4.1
r37_feynman_shared_var	Shared-variable remap (ℓ'=0.00, n=25)	§4.1
r39b/d/f_leakage_vs_scale	Augmentation-volume curve (Fig. 6)	§M
r40_family_scramble	Token-similarity alignment (App. E, n=25)	App. E
r41_head_robustness	Evaluation-head robustness (App. J, n=25)	App. J
r42_triplet_only_mlp	Triplet-only MLP identification (n=10)	§4
r44_lample_charton_encoder	Lample-Charton trained encoder audit (n=25)	App. B
r45_triplet_only_e3b	Triplet-only under E3b protocol (n=25)	App. L
r46_bce_mechanism	BCE mechanism λ-sweep (n=10)	App. R
r47_equation_level_stats	Equation-level bootstrap analysis	§4.5
r48_triplet_only_shared_var	Triplet-only shared-variable, identity-renaming (superseded by r52)	§4.1
r49_triplet_only_mlp_n25	Triplet-only MLP (0.77, n=25)	§4
r50_bce_mechanism_mlp	BCE mechanism λ-sweep with MLP (n=10)	App. R
r51_r22_equation_bootstrap	Equation-level OOL CI (from r22)	§4.5
r52_shared_var_permutation_rename	Shared-var, genuine within-range renaming (0.93, $\ell'\approx$0, n=25)	§4.1
r53_shared_var_op_scramble	Shared-var operator-scramble audit (0.93→0.28, n=25)	§4.1
r54_shared_var_arity_controlled	Shared-var arity-controlled audit (0.89 vs 0.17 chance)	§4.1
r56_overlap_extended	12-law leakage-proof suite (0.61 vs 0.46 random, eq-level CI; n=25)	§P
r55_benchmark_survey	*superseded by* r57 (adds EQNET and the S2 column)	—
r57_eqnet_survey	Unified separability survey, 23 of Table 23's 25 corpora	§4.1
r58_eqnet_diagnostic	Full diagnostic on EQNET (Allamanis et al., 2017), their splits	§4.1
r59_leave_one_schema_out	Leave-one-schema-out coverage manipulation (TED held constant)	§4.5
r60_separability_predicts_skyline	Screening prediction across 12 corpora (negative result)	§S
r61_equation_level_positive	Equation-level CIs for the positive control and arity control	§4.1
r62_nesymres_global_permutation	Global-permutation external control on NeSymReS (null)	§4.1
r63_matched_contrast	Table 21; supersedes r21/r33 pairing	§4.5
r64_overlap12_tier2	Overlap-12 bag/scramble skylines (retracts the 12-law claim)	§P
r66_eqnet_duplicate_check	EQNET held-out forms vs training forms (0/747 identical)	§4.1
r67_score5_skyline	*superseded by* r68 (non-matching protocol)	App. T
r68_score5_exact	Table 20: floor under EqNet's own protocol	§4.1
r90_external_audit_lample	Table 31: the level ladder on an external benchmark and an external model (n=25); extends r44	§4.2
r28_external_audit_nesymres	*superseded by* r62 (its control permuted held-out sets only)	—

L. TRIPLET-ONLY UNDER E3B (OUT-OF-LIBRARY TRANSFER)

The triplet-only encoder reaches 0.77 on the 5-law overlap suite. §P withdraws that as evidence of structural identification (a variable-blind bag scores 0.824 on the same suite) but the transfer question below is still worth asking of the same encoder, and its answer does not depend on the withdrawn claim. Does this structure transfer to out-of-library held-out forms? We train a triplet-only encoder on the Feynman corpus (same protocol as r42) and evaluate under the E3b protocol (n=25 seeds; tag `r45_triplet_only_e3b`).

Result: triplet-only E3b plain $= 0.52 \pm 0.05$, renamed $= 0.08 \pm 0.07$; joint E3b plain $= 0.53 \pm 0.05$, renamed $= 0.08 \pm 0.08$ (chance $= 0.10$). The two objectives are indistinguishable under E3b ($t = -0.59$, p=0.56). The structural identification that triplet-only achieves in-distribution (on leakage-proof laws with shared variables) does *not* transfer to out-of-library forms. This cleanly separates two findings: (1) triplet-only identifies structure when the benchmark design permits it (shared variables, in-distribution); (2) no training objective tested, including triplet-only, generalizes structurally to forms outside the training library on the Feynman benchmark.

This result strengthens rather than weakens the paper's constructive recommendation: proper benchmark design (shared variables) is necessary for structural learning to *manifest*; the transfer failure is a property of the benchmark's equation diversity (10 equations with distinct variable subsets) rather than the training objective.

Provenance. Code: `run_triplet_only_e3b.py`. Data: `logs/r45_triplet_only_e3b.json`.

M. LEAKAGE IS NOT A SMALL-DATA OR TRAINING-STRENGTH ARTEFACT

Augmentation-volume robustness. To test whether leakage is a small-data artefact that vanishes with more training data, we vary *augmentation volume*, paraphrases per equation, while holding equation diversity fixed at 10 Feynman equations under the E3b (out-of-library) protocol. Tree-LSTM encoders are trained at four volume points: 500, 2,000, 8,000, and 20,000 training expressions (n=10, 5, 5, 25 seeds respectively; bootstrap 95% CIs; Figure 6). Results: ℓ'=0.78 [0.68, 0.92] at 500; 0.82 [0.70, 0.92] at 2K; 0.78 [0.67, 0.89] at 8K; 0.75 [0.64, 0.85] at 20K.[1] Leakage remains high ($\ell' \geq 0.75$) at every volume tested. We do not fit a slope because the between-run-family spread ($\sim$0.10 between tags `r34` and `r39`) exceeds any volume effect over the $40\times$ range; the data support a *level* claim, not a *trend* claim. This rules out training insufficiency as an explanation *within fixed-diversity regimes*: more gradient exposure to paraphrases of the same 10 equations does not teach variable-identity invariance. The meaningful "scale" dimensions, equation diversity and model capacity, require architectures beyond scope, and **we offer no external evidence on them**: the NeSymReS audit (Appendix I, 10M-param Transformer trained on 100+ equations) returns a *null* under the correct global-permutation control (tag `r62`), so it establishes that the diagnostic *ports* to such a model and nothing about whether leakage survives scale. **This paragraph's conclusion therefore rests on the fixed-diversity curve above and the hardened recipe below, not on any external model.** A hardened training recipe (§H) further confirms this at the recipe level: near-perfect in-library accuracy (0.992) coexists with ℓ'=0.99 out-of-library.

Hardened training recipe. To rule out the objection that our baseline training is too weak to learn invariant features, we train the strongest recipe available on the Feynman corpus: k=8 chained paraphrases (depth 4, composing 4 sequential rewrites drawn from a 13-rewrite list that includes full random-permutation alpha renaming as one option; at each step a random rewrite targets a random subtree, so alpha is applied with probability $\approx 1/13$ per step), 20 epochs (n=25 seeds; Appendix H). This produces structurally distant positives with low token overlap, explicitly forcing the triplet objective beyond surface cues. Results: E3 (in-library) plain $= 0.992 \pm 0.040$, ℓ'=0.37. Training succeeds (near-perfect plain accuracy), but in-library leakage *increases* relative to the baseline (ℓ' rises from 0.17 to 0.37): since most chain steps apply structural rewrites (commutativity, distribution, exp/log) that preserve variable indices, most positives share substantial variable-index overlap

[1] The 500-expression point (ℓ'=0.78, n=10, tag `r39b`) differs from Table 21's E3b (ℓ'=0.88, n=25, tag `r26`) because the curve uses fewer seeds and a separate run family; the 95% CIs overlap $[0.68, 0.92]$ vs $[0.80, 0.94]$.

with the anchor, and the triplet loss learns more refined variable-pattern features to discriminate structurally similar but variable-distinct negatives. E3b (out-of-library) plain $= 0.480 \pm 0.071$, ℓ'=0.99 $[0.95, 1.00]$: leakage is even higher than the baseline (ℓ'=0.88). Stronger training amplifies the variable-identity shortcut in-library while transferring no structural knowledge to out-of-library forms. This is not a training-insufficiency artefact: the encoder demonstrably learns (E3 near-perfect) yet the learned representations remain distribution-specific.

N. Disclosure vs. Fix: Full Taxonomy

Not all responses to variable-identity leakage are equivalent. We distinguish three categories, ordered by the strength of the guarantee they provide:

- A *disclosure* technique removes the leakage signal from the encoder (collapsing performance to the random-encoder level) but does not restore structural identification.
- A *within-library fix* restores above-chance structural identification within the training distribution, i.e. under variable renamings the encoder was trained to be invariant to.
- A *transfer fix* would additionally hold on out-of-library algebraic forms: the encoder generalises its structural invariance beyond the training distribution.

De Bruijn canonicalization is a disclosure technique; alpha-renaming augmentation is a within-library fix but *not* a transfer fix, as we establish below.

De Bruijn canonicalization (E2): replacing variable indices with canonical de Bruijn positional indices before tokenization eliminates variable-identity information at the input. Result: within-range diagonal-correctness 0.210 ± 0.092, equivalent to the random encoder's 0.200 ± 0.079 by TOST (p=0.0002). Leakage is gone, but so is any trained advantage: the encoded representations carry no variable-identity signal, and the encoder has not learned to substitute anything structural in its place.

Alpha-renaming augmentation: adding random cyclic shifts of variable indices within $\{0, \ldots, 3\}$ to the paraphrase library forces the encoder to learn variable-independent features. All three encoders are re-trained from scratch (25 seeds). Under within-range renaming, the alpha-augmented encoder achieves 0.330 ± 0.162 versus baseline 0.170 ± 0.093 and random 0.180 ± 0.062 (p=0.0002, d=0.90): $2.6\times$ chance vs $1.4\times$ for baseline (Table 17). A practitioner who applies de Bruijn to "solve" the leakage problem has disclosed it, not fixed it. Alpha-renaming is the within-library fix.

Transfer check: alpha-augmented encoder on out-of-library forms. Does the within-library fix transfer to out-of-library forms? We train the alpha-augmented encoder and evaluate under both E3 and E3b protocols (25 seeds; Appendix F). Result: E3-plain $= 0.95 \pm 0.08$ (encoder works in-library), E3b-plain $= 0.12 \pm 0.09$ (collapses on OOL forms), a genuine *transfer failure*, not a training failure. *Unified account of OOL plain accuracies across recipes:* three training regimes produce qualitatively different OOL outcomes: alpha-only collapses (0.12), baseline is near-chance (0.50), hardened recipe is also near-chance (0.48). Alpha-only augmentation uses cyclic shifts within a 4-variable vocabulary: too narrow for the encoder to learn genuine invariance; instead it memorizes the shift pattern itself, and OOL forms that violate this pattern produce incoherent embeddings. The hardened recipe (chained rewrites with alpha as one of 13 possible rewrites per step) generates structurally diverse positives that receive alpha renaming on at most 1–2 of 4 chain steps (probability $\approx 1/13$ per step) and typically on subtrees rather than the full expression, so most positives share substantial variable-index overlap with the anchor; the encoder neither collapses nor generalises, and OOL accuracy matches the no-augmentation baseline ($0.50 \approx 0.48$; ℓ'=0.99). The common thread is that none of the three recipes transfers structural knowledge to OOL forms: they differ only in whether the training distribution is narrow enough to cause catastrophic collapse. Together these confirm alpha-renaming is a within-library fix only. Variable-identity tokens are the dominant discriminative cue in this small-data regime (Appendices E, B). Multi-form robustness: 50 schema-generated OOL forms across all 10 Feynman equations yield ℓ' mean $= 0.778$, range $[0.647, 0.926]$ (25 seeds per form; Appendix F). An independent evaluation using practitioner-style algebraic rewrites (SymPy factor/apart/explicit-reciprocal, product reassociation; 12 forms across 8 equations after SymPy structural-identity deduplication; n=25 seeds each) yields ℓ' mean $= 0.800$, range $[0.705, 0.885]$: high leakage is robust across both mechanically- and naturally-generated OOL

forms (Appendix F). The contrast confirms on all 10 equations as well as on the TED-cleaned 8 (Table 21).

O. Supporting Analyses for the Coverage Section

Chance correction. Because different protocols evaluate different equation subsets (chance $c{=}1/K$), the raw ℓ values are not directly comparable. The chance-corrected index $\ell' = 1 - (\text{renamed} - c)/(\text{plain} - c)$ normalises above-chance performance and is comparable across protocols. The robust contrast is the $5\times$ swing between E3 and E3b: $\ell'{=}0.17$ $[0.13, 0.21]$ vs $\ell'{=}0.88$ $[0.80, 0.94]$, both computed over all 10 equations with stable per-equation behavior.

Ratio-form stability. Because ℓ' has a ratio form, small denominators (plain $- c$) could inflate it near chance. A simulation study confirms it recovers a known sensitivity whenever denom/SE > 5, which both protocols in Table 21 satisfy (83.6 and 12.6). The secondary metric (renamed $- c$) preserves the same ordering, so the contrast is not an artefact of the ratio form, but the (plain, renamed) pairs are the evidence regardless.

Token-similarity alignment check (E3b). A token-similarity oracle that assigns each E3b held-out form to the most bag-similar centroid achieves 0.90 accuracy (9/10), and the trained encoder's E3b predictions align with the oracle significantly more than independence predicts (observed agreement $= 0.604$, expected under independence $= 0.459$; $t{=}8.15$, $p{<}0.0001$, $n{=}25$; Appendix E). Token co-occurrence patterns contribute materially to E3b residual identification. The 0.148 E3b trained renamed is significantly above chance ($p{=}0.011$) but does not constitute a structural residual: an arity-aware bag classifier scores 0.375 under the same renaming, and a weighted within-arity-group analysis (2-variable group: 6 eqs, chance $= 0.167$, encoder $= 0.067$; 1-variable group: 2 eqs, chance $= 0.500$, encoder $= 0.440$; weighted encoder $= (6{\times}0.067 + 2{\times}0.440)/8 = 0.160$; weighted chance $= 0.250$) shows the trained encoder below within-group chance (Appendix E).

E3 residual explanation. The 8-equation E3 retains renamed accuracy 0.920 ($7.4\times$ the 8-way chance of 0.125). This does not constitute structural learning: among the 8 included equations, 8/8 have unique (arity, tree-depth, operator-bag) signatures. Renaming permutes only *which* variable indices appear, not the number of variables, tree depth, or operator types, so the encoder can discriminate equations by structural fingerprint alone after renaming. Arity is the dominant cue: the two arity-1 equations achieve 0.98 renamed accuracy (perfectly distinguishable from all arity$\geq$2 equations), while the six arity-2 equations average 0.87 (confusion only within the same arity group). The residual thus reflects the leniency of the E3 protocol, whose in-library held-out forms are single-rewrite paraphrases preserving all structural cues except variable identity.

Clustered correlation. The within-OOL $\rho{=}{-}0.13$ (n.s., $n{=}50$ forms) treats 5 forms per equation as independent. Accounting for within-equation clustering reveals a Simpson's-paradox structure: the equation-level correlation is $\rho{=}{-}0.81$ ($p{=}0.004$, $n{=}10$ equations), but the residualized within-equation correlation is $\rho{=}{+}0.52$ ($p{<}0.001$). The negative between-equation effect is an arity/operator-diversity confound (equations with more complex OOL forms also have more distinctive structural signatures, lowering ℓ'), not a causal effect of structural distance on leakage. The positive within-equation effect (larger TED $\rightarrow$ slightly higher ℓ') is modest and consistent with the boundary interpretation: more distant OOL forms are less likely to accidentally share structural features with the library. Critically, even the most TED-proximal OOL forms (TED$= 5$) show $\ell' > 0.64$: the boundary, not distance, remains the operative predictor.

Dual-skyline divergence. The two standard non-learned baselines measure *different things*. In-distribution (E3), bag 0.992 and random encoder 0.724 are both driven by variable tokens, and diverge only modestly. It is worth being careful about what the random encoder's score licenses: a randomly initialised recursive network is *not* a random projection of the token multiset (its output geometry carries tree topology, depth and nonlinear interactions as well) so the right reading is that **random initialization alone can produce highly discriminative geometry**, i.e. embedding-based retrieval inherits substantial structure from the architecture and input representation themselves, rather than that token geometry already reflects algebraic proximity. Under distribution shift (E3b), they diverge $7\times$: bag 0.900 vs random encoder 0.124 on identical forms with identical centroid

protocol. The bag accumulates token counts regardless of tree structure; the random encoder's recursive nonlinearity decouples from the bag when held-out tree shapes differ from training tree shapes. Practitioners should report both skylines: when they diverge sharply, a distribution shift is present between training and held-out forms, and the random-encoder skyline understates the token-bag ceiling. The gap is itself a distribution-shift diagnostic (Appendix C).

Table 11: **Revision history: seven incidents, how each was caught, and what stands now.** Every one was found by a check this paper prescribes, which is the calibration behind the claim that the protocol is sensitive enough to catch its authors' own errors.

Incident	What was wrong	How it was found	What is in the paper now
`r17` phantom number	A reported alpha-augmented E3b renamed accuracy of 0.212 ± 0.098 had never been computed; it was carried over from an intermediate annotation.	Log audit: no run tag produced the value.	The first genuine run (25 seeds, tag below) gives renamed $= 0.104 \pm 0.082$, p=0.81; every number now cites a provenance tag.
`r20` vacuous held-out	One doppler-factor OOL form was algebraically identical to the hand-crafted held-out, so that comparison compared a form with itself.	Equivalence check against the held-out, before submission.	Retracted; `r22` deduplicates syntactically against canonical *and* held-out forms.
`r26` deduplication gap	The same recurrence: SymPy's `factor` form of that equation matched the held-out structurally while differing as a string.	The string-distinctness check passed; no structural check had been run.	A SymPy structural-identity check excludes it, $13 \to 12$ forms.
`r17`/`r18` zero triplet loss	Zero triplet loss was read as evidence of representation collapse.	The E3 diagnostic contradicted it: E3-plain $= 0.95$ under zero loss.	Reinterpreted as in-distribution memorisation; the failure is one of *transfer* (E3b-plain $= 0.12$).
Tree-edit distance defect	The post-order routine returned each node's own index in place of its left-most leaf, so $l(i)$ never propagated up a left spine. Both the keyroot set and the forest recurrence were then computed from a wrong $l()$, changing the distance on a majority of pairs.	**Running the ladder on an external corpus** (§4.2), whose trees are deeper than any of ours, raised `IndexError`. Checked against an independently written Zhang–Shasha reference: the fix agrees on $3000/3000$ sampled pairs, the shipped code on 453.	Fixed. The swap-twin baseline moves $0.733 \to 0.723$ and the rotation twin to 0.889: **both corrected values make the baseline *weaker***, so the correction runs in this paper's favour, which is stated rather than left implicit.
`r91` alternate-order aggregation	Alternate-order scores were accumulated into a per-*ladder* list from inside a loop over training *libraries*, so the arm averaged the full and leave-one-primitive-out libraries while the forward score it was subtracted from was the full library alone. A $+0.250$ "order cost" was a contrast between two *training conditions*.	**Running the experiment the appendix admitted was missing** (AV): the port reproduced `r91`'s forward and alternate *cells* within seed noise but returned $+0.029$ for its *delta*. A discrepancy in neither operand localises the fault to the aggregation; all six published deltas then matched $\text{fwd}_{\text{full}} - \overline{\text{alt}}_{\text{pooled}}$ exactly.	**Withdrawn.** Within a fixed library no order cost exceeds 0.035 and four of twelve are negative, so all six stored "excludes zero" flags were spurious. Table 32's row is corrected and the verifier now recomputes the decomposition. **This one runs *against* an earlier caveat of ours**, removing a scope condition rather than adding one.
NeSymReS control arm	Appendix I reported ℓ'=0.44 for a published external encoder from a control permuting the *held-out* sets alone, which breaks the centroid–query correspondence and so measures the control, not the encoder.	**The run index recorded `r28` as superseded by `r62`** while the prose that used it, 100 lines away, still cited `r28`; `verify_claims.py` had *no* NeSymReS assertion at all.	**Withdrawn** (I). Under one global permutation, which leaves the corpus isomorphic, $\Delta = +0.024$ $[-0.036, 0.084]$: a null. The verifier now asserts both arms.

Table 12: TED vs. ℓ': summary statistics by protocol cluster. E3m forms are in-library (same paraphrase library) but use exp/log/power rewrites, showing high TED overlapping the OOL range. **The E3m row (n=4) is an unstable probe**, see *E3m instability* below, **and no claim in the body rests on it.**

Protocol cluster	Membership	n	$\text{TED}_{\min}$ (mean [range])	ℓ' (mean)
E3 library	in-library	10	2.7 [0, 5]	0.17
E3m exp/log rewrites	in-library	4	6.3 [3, 8]	0.41
E3b hand-crafted	OOL	10	6.1 [3, 8]	0.88
r22 schema	OOL	50	7.5 [5, 18]	0.78

Table 13: Cluster contrast and within-OOL correlation for $\text{TED}_{\min}$ vs. ℓ'. The Mann-Whitney test shows that in-library and OOL TED distributions are well separated, but TED does not predict ℓ' within OOL: the training-coverage boundary is the operative predictor.

Test	Subset	Statistic	p-value
Mann-Whitney U (TED: in-lib vs. OOL)	n=10 vs n=60	U=50.5, z=−4.19	< 0.001
Spearman ρ (TED vs. ℓ', within OOL)	n=50	ρ=−0.13	n.s.
Pearson r (TED vs. ℓ', within OOL)	n=50	r=−0.06	n.s.

Table 14: Full TED analysis data (tag `r23_tree_edit_distance`). $\text{TED}_{\min}$ = minimum edit distance to nearest training form (canonical + 4 paraphrases, seed 0).

Equation	Protocol	Form	$\text{TED}_{\min}$	ℓ'
gaussian	E3 library	—	5	0.17
coulomb_force	E3 library	—	2	0.17
grav_pe	E3 library	—	2	0.17
spring_pe	E3 library	—	5	0.17
capacitor_v	E3 library	—	3	0.17
wavenumber	E3 library	—	0	0.17
cyclotron	E3 library	—	3	0.17
thermal_energy	E3 library	—	2	0.17
energy_density	E3 library	—	5	0.17
doppler_factor	E3 library	—	0	0.17
gaussian	E3m in-library	exp/log rewrite	7	0.41
spring_pe	E3m in-library	exp/log rewrite	8	0.41
energy_density	E3m in-library	exp/log rewrite	3	0.41
doppler_factor	E3m in-library	exp/log rewrite	7	0.41
gaussian	E3b hand-crafted	—	7	0.88
coulomb_force	E3b hand-crafted	—	7	0.88
grav_pe	E3b hand-crafted	—	8	0.88
spring_pe	E3b hand-crafted	—	8	0.88
capacitor_v	E3b hand-crafted	—	4	0.88
wavenumber	E3b hand-crafted	—	6	0.88
cyclotron	E3b hand-crafted	—	8	0.88
thermal_energy	E3b hand-crafted	—	3	0.88
energy_density	E3b hand-crafted	—	5	0.88
doppler_factor	E3b hand-crafted	—	5	0.88

Table 15: Full TED analysis data, continued: the 50 `r22_schema` multi-form points (tag `r23_tree_edit_distance`). Columns as in Table 14.

Equation	Protocol	Form	$\mathrm{TED}_{\min}$	ℓ'
gaussian	r22 schema	0	18	0.873
gaussian	r22 schema	1	13	0.705
gaussian	r22 schema	2	11	0.705
gaussian	r22 schema	3	5	0.705
gaussian	r22 schema	4	7	0.705
coulomb_force	r22 schema	0	5	0.831
coulomb_force	r22 schema	1	5	0.862
coulomb_force	r22 schema	2	5	0.880
coulomb_force	r22 schema	3	5	0.888
coulomb_force	r22 schema	4	5	0.892
grav_pe	r22 schema	0	12	0.887
grav_pe	r22 schema	1	9	0.738
grav_pe	r22 schema	2	5	0.706
grav_pe	r22 schema	3	5	0.706
grav_pe	r22 schema	4	5	0.738
spring_pe	r22 schema	0	14	0.667
spring_pe	r22 schema	1	12	0.657
spring_pe	r22 schema	2	5	0.647
spring_pe	r22 schema	3	9	0.647
spring_pe	r22 schema	4	10	0.667
capacitor_v	r22 schema	0	7	0.912
capacitor_v	r22 schema	1	5	0.853
capacitor_v	r22 schema	2	5	0.833
capacitor_v	r22 schema	3	5	0.922
capacitor_v	r22 schema	4	5	0.926
wavenumber	r22 schema	0	7	0.838
wavenumber	r22 schema	1	5	0.762
wavenumber	r22 schema	2	5	0.808
wavenumber	r22 schema	3	5	0.882
wavenumber	r22 schema	4	5	0.857
cyclotron	r22 schema	0	15	0.876
cyclotron	r22 schema	1	11	0.683
cyclotron	r22 schema	2	7	0.683
cyclotron	r22 schema	3	5	0.683
cyclotron	r22 schema	4	5	0.683
thermal_energy	r22 schema	0	12	0.886
thermal_energy	r22 schema	1	5	0.736
thermal_energy	r22 schema	2	9	0.816
thermal_energy	r22 schema	3	5	0.704
thermal_energy	r22 schema	4	5	0.704
energy_density	r22 schema	0	9	0.846
energy_density	r22 schema	1	9	0.846
energy_density	r22 schema	2	10	0.846
energy_density	r22 schema	3	5	0.846
energy_density	r22 schema	4	5	0.846
doppler_factor	r22 schema	0	13	0.714
doppler_factor	r22 schema	1	5	0.706
doppler_factor	r22 schema	2	10	0.706
doppler_factor	r22 schema	3	5	0.706
doppler_factor	r22 schema	4	7	0.706

Table 16: Per-equation ℓ' for E3 and E3m protocols. E3: 10-way, n=25 seeds, c=0.10. E3m: 4-way over 4 eligible equations, n=25 seeds, c=0.25. †: raw $\ell' > 1$ (renamed below chance), clipped to 1 per §3. Negative values indicate renamed above plain (noise-dominated at single-equation level). Dashes: equation not in E3m subset, or undefined (plain = 0).

	E3 (in-library)			E3m (held-out family)		
Equation	plain	ren	ℓ'	plain	ren	ℓ'
gaussian	1.000	1.000	0.000	1.000	0.960	0.053
coulomb	1.000	0.840	0.178	—	—	—
grav_pe	1.000	0.920	0.089	—	—	—
spring_pe	1.000	0.920	0.089	0.960	0.120	1.00†
capacitor	0.960	0.840	0.140	—	—	—
wavenumber	0.960	0.000	1.116	—	—	—
cyclotron	1.000	1.000	0.000	—	—	—
thermal	1.000	0.880	0.133	—	—	—
energy_density	0.920	0.960	−0.049	0.600	0.840	−0.686
doppler	0.880	0.880	0.000	0.000	0.000	—
Aggregate	0.972	0.824	0.17	0.643	0.480	0.41

Table 17: Disclosure vs. fix (separate run family; all comparisons within-family). Mean ± std, 25 seeds. Within-range = leakage-free; chance $1/8$=0.125 (main) or $1/5$=0.20 (overlap).

Encoder	Main plain diag.	Main within-range diag.	Overlap plain diag.
Alpha-augmented Tree-LSTM	0.805 ± 0.094	0.330 ± 0.162	0.35 ± 0.13
Baseline Tree-LSTM	0.845 ± 0.081	0.170 ± 0.093	0.33 ± 0.13
De Bruijn Tree-LSTM	0.855 ± 0.068	0.210 ± 0.092	—
Random encoder	0.430 ± 0.062	0.180 ± 0.062	0.40 ± 0.06

P. The Overlap Suite Under the Joint Objective

Q. Domain Guards and Sampling

Domain guards are enforced during evaluation: division clamps the denominator to $\max(|\text{denom}|, 10^{-6})$, logarithm operates on $\log(|x| + 10^{-6})$, square root on $\sqrt{|x|}$, and exponential output is clipped to $[-20, 20]$. For each expression, we attempt to sample satisfying and non-satisfying assignments via a mixed random and hard-negative strategy with retry budget $k = 64$. Expressions with no satisfying assignment after k retries are discarded.

Equivalence classes could in principle be merged by numerical signature (expressions with identical satisfaction patterns over a canonical assignment set). We evaluated and rejected this: the post-filter false-merge rate was 13% (100 pairs, 87 verified equivalent by SymPy, 13 refuted). Expressions are filtered by satisfaction rate to $[0.1, 0.9]$, since 46.8% of unfiltered expressions are near-constant over the sampling range and provide no training signal.

Table 18: Variable-leakage-proof overlap laws: five laws all over variables x_0, x_1, under the *joint* triplet+BCE objective with the jointly-trained readout head. Mean $\pm$ std across n=25 seeds (Tree-LSTM, Random) or n=5 (GIN; directional only); chance $= 0.20$. The random encoder's zero variance is not an anomaly: two of the five laws are operator-homogeneous and are separated under *any* projection, so every seed scores exactly $2/5$. Triplet-only results, which are the ones that clear this floor, are in Table 19.

Encoder	Diagonal-correctness
Tree-LSTM (joint)	0.32 ± 0.098
GIN (joint)	0.24 ± 0.150
Random (untrained, deterministic)	0.40 ± 0.000

R. Objective $\times$ Evaluation-Head Factorial on the Overlap Suite

Table 19: Evaluation-method sensitivity on the leakage-proof overlap suite (5 laws, shared variables x_0, x_1; chance $= 0.20$). The joint condition produces opposite orderings under MLP-based vs. centroid-based identification; triplet-only helps under both. MLP results: n=25 seeds, all conditions (tag `r49` for triplet-only). Centroid results: n=25 seeds, 4 conditions. The joint-MLP row reports 0.32 ± 0.12 where Table 18 reports 0.32 ± 0.098: these are two independent run families evaluating the same condition; the means agree, and the difference is seed-sampling spread in the standard deviation, not a discrepancy in the estimate. As in Table 18, the deterministic 0.40 ± 0.00 rows arise because two of the five laws are operator-homogeneous and separate under any projection.

	MLP-based (physics predictor)		Centroid-based (nearest-mean)	
Condition	Overlap acc.	vs. random	Overlap acc.	vs. random
Joint (triplet+BCE)	0.32 ± 0.12	-0.08 (p=0.017)	0.66 ± 0.09	$+0.08$ (p<0.001)
BCE-only	0.40 ± 0.00	0.00	0.53 ± 0.10	-0.06 (p<0.001)
Triplet-only	0.77 ± 0.14	$+0.37$ (p<0.001)	0.70 ± 0.11	$+0.11$ (p<0.001)
Random encoder	0.40 ± 0.00	—	0.58 ± 0.05	—

S. Screening, and the Released Command-Line Auditor

A separability statistic is only useful if it tells a practitioner something they would otherwise have to train a model to learn. We therefore stated a prediction with no free parameters (an untrained recursive encoder is, to first order, a random projection of the token multiset, so its identification accuracy should track token-bag separability) and tested it across 12 corpora (5 ours, 7 EQNET) under one uniform held-out-form protocol (tag `r60`, n=25 untrained encoders per corpus).

The prediction fails, and part of that is our own fault. Measured against the identity line, pairwise token-bag separability gives $R^2 = -10.6$ and the zero-parameter bag *classifier*'s own accuracy gives $R^2 = -2.8$, with a correlation to untrained-encoder accuracy that is if anything negative ($r = -0.32$, n=12, an interval wide enough that we draw no conclusion from the sign). The first number is largely an artefact of our own choice of statistic: pairwise separability sits in $[0.92, 1.00]$ on every corpus we measured, and a predictor with no variance cannot predict. Restriction of range guarantees the failure, so we report the bag-accuracy version as the informative one and do not claim the pairwise result as evidence of anything beyond saturation.

What remains after that concession is still a negative result, and it has teeth. Bag accuracy and untrained-encoder accuracy measure genuinely different things: on EQNET ONEVARPOLY13 the bag scores 0.15 where the untrained encoder scores 0.66, and on our Feynman-10 the ordering reverses (1.00 versus 0.13). A recursive nonlinearity is sensitive to tree shape a bag discards, and blind to count differences a bag sees. So we could not find a training-free statistic that predicts what an untrained model will score, and the checklist's stage 2 (report *both* non-learned baselines, because neither substitutes for the other) is the operational consequence. The audit establishes what a shortcut *could* exploit, and at S1 whether the benchmark is a lookup table by construction; it does not forecast a number. A margin-based statistic that does not saturate might do better, and we leave that open rather than claiming the question is closed.

The auditor, released. Stage 1 of the checklist requires no training and is shipped as a dependency-light command-line tool: `audit-symbolic-benchmark corpus.json` reports unique var-set fraction, per-level pairwise separability, arity distribution, and the (appropriately caveated) projected skyline, with adapters for all corpora in Table 23 and a documented input format for new ones.

T. The Superseded score_5 Comparison, and Why It Was Wrong

Superseded, and kept only so the corrected comparison can be checked against the one it replaced. Appendix V carries the figures the paper uses; nothing here bears on a current claim. Our first attempt at the leaderboard comparison of Table 20 (tag `r67`) computed score_5 with the UNSEENEQCLASS split serving as its own neighbour pool, and normalised by k. Under that procedure the structure-free floor looked dramatic: a bag or an untrained encoder matched or beat the

StructEmb ablation on 13 of 14 corpora, beat the headline SemEmb system on 3, and reached 69.4 on BOOL8 where EqNet reports 58.1.

It was wrong, and the direction of the error was not random; every difference from the original protocol made our numbers larger:

1. *Neighbour pool.* Allamanis et al. (2017) embed the entire corpus and search neighbours in it; the test split only supplies queries. On BOOL8 that is $257{,}784$ candidate neighbours rather than $13{,}859$. More distractors can only lower a score.
2. *Normalisation.* Their score divides by $\min(|\text{class}|, k)$, the number of same-class neighbours actually achievable, not by k.
3. *Query set.* Their queries are corpus entries whose token string appears in the test split, so each query sits in the pool alongside the rest of its class.

Reproducing all three (tag `r68`) moves the comparison from "matches or beats StructEmb on 13 of 14" to 5 of 14, and from "beats SemEmb on 3" to none. We record the superseded numbers here rather than deleting them, because the failure mode is one this paper is otherwise about: we had reported both non-learned baselines under a single internally consistent protocol, satisfying our own checklist, and still produced a misleading comparison, because the protocol was ours and the numbers beside it were someone else's.

U. EVERY CLAIM AGAINST ITS STRONGEST SKYLINE

Four times in this revision, supplying a non-learned baseline we had not measured reversed a conclusion we had drawn: the 12-law suite, the 5-law suite, the score_5 leaderboard, and the interpretation of Table 21. Each time we had fixed the instance in front of us without sweeping for the class. Tag `r69` is the sweep: for every protocol under which this paper reports an identification number, it measures all three bag representations and an untrained encoder *under that same protocol*, $n{=}25$ seeds.

Protocol	K	full-token	var-blind	var-only	untrained	trained encoder
Matched contrast, in-library	8	0.980	0.715	**1.000**	0.790	0.965
Matched contrast, out-of-library	8	**1.000**	0.600	**1.000**	0.435	0.832
Feynman E3 (10 equations)	10	0.932	0.636	**1.000**	0.748	0.972
Shared-variable Feynman	10	0.624	0.564	0.496	0.628	**0.932**
EQNET SIMPLEPOLY5	38	0.368	0.105	0.342	0.289	**0.944**
EQNET SIMPLEPOLY8	90	0.467	0.056	0.244	0.407	**1.000**

Read down the boldface. On the three physics-derived protocols a zero-parameter bag over variable tokens alone is perfect or near-perfect, and the trained encoder is at or below it: these protocols cannot license a claim about structure, which is the paper's thesis applied to its own measurements. Only the shared-variable Feynman suite and the two EQNET corpora have a trained encoder above every non-learned baseline, and those are the only identification claims we advance. The EQNET rows use the sweep's uniform protocol (up to 90 classes, one held-out form per class) and so differ from the numbers in §4.1, which follow EQNET's own splits; both are reported rather than reconciled into one, since they answer different questions.

V. THE score_5 LEADERBOARD FLOOR, IN FULL

Table 20: **A zero-parameter floor under a published leaderboard, computed under that leaderboard's own protocol** (tag `r68`). score_5 (%) on the UNSEENEQCLASS split of the 14 SemVec corpora. The right-hand columns are transcribed from the published papers: EqNet (Allamanis et al., 2017), and the StructEmb autoencoder ablation and headline SemEmb system of Gangwar & Kani (2023). The left-hand columns are non-learned representations we compute: a token-multiset bag, a variable-blind bag (operators, constants and arity), and an untrained Tree-LSTM (n=3 initialisations, over the full pool in every case). Nothing on the left is trained. Crucially, all of it is evaluated exactly as Allamanis et al. (2017) evaluate: neighbours are searched in the *full corpus* (*pool*), not in the test split; cosine distance; self excluded; and the score normalised by $\min(|\text{class}|, 5)$ rather than by 5. **No non-learned representation reaches EqNet or SemEmb on any corpus.** Boldface marks the five where one reaches the StructEmb ablation.

			non-learned (this work)			*published*		
Corpus	Pool	Queries	Bag	Var-blind	Untrained	EqNet	StructEmb	SemEmb
Bool8	257784	13859	12.7	2.6	**31.6**	58.1	30.6	100.0
oneV-Poly13	107725	8032	3.7	2.6	32.9	90.4	38.7	99.6
SimpPoly10	57909	6189	31.8	8.2	36.1	99.3	40.1	99.8
SimpBool8	39048	3096	9.5	5.7	**44.5**	97.4	36.9	99.4
Bool10	51200	8600	4.8	2.1	**14.4**	71.4	10.8	91.3
SimpBool10	26304	4536	6.6	2.0	**37.9**	99.1	24.4	95.5
BoolL5	36050	4656	30.3	0.2	22.3	75.2	38.3	36.0
Poly8	11451	1234	10.6	0.6	26.2	86.2	32.7	87.3
SimpPoly8	3477	362	38.5	2.2	31.9	98.9	47.6	98.9
SimpBoolL5	10050	1242	51.5	0.0	23.1	85.0	55.1	71.1
oneV-Poly10	1291	102	44.1	27.5	47.2	81.3	59.8	74.1
Bool5	1239	154	34.7	11.6	22.9	65.8	36.4	57.7
Poly5	516	74	**41.8**	7.0	31.5	55.3	5.7	45.2
SimpPoly5	237	24	9.4	0.0	11.1	65.6	28.1	20.8

W. TRAINING-LIBRARY COVERAGE: FULL LOSO DESIGN AND THE MATCHED CONTRAST

A leave-one-schema-out design identifies a *causal* effect of library membership under a matched-schema intervention (tag `r59`); the causal reading is licensed by the intervention, and we do not extend it to observational coverage. For each of eight out-of-library rewrite schemas we train two encoders whose held-out trees are *byte-identical*, varying only whether that schema is in the training paraphrase library. Five schemas are at ceiling in the OUT arm; on the three with headroom the effect is unambiguous ($\Delta\text{plain} = +0.170, +0.170, +0.150$, all with intervals excluding zero), pooling to $+0.061\ [0.019, 0.123]$. Library membership *per se* raises identification with structural distance held fixed. It is a supported mechanism, not the sole explanation: the effect is $\approx+0.16$ where headroom exists against a -0.13 drop across the two arms of Table 21 (details and the full schema breakdown in Appendix O).

Table 21 is the paper's single quoted contrast: holding equation set, encoder, procedure and run family fixed, the (plain, renamed) pair moves from $(0.965, 0.880)$ to $(0.832, 0.662)$. Figure 4 is that contrast as panel B, beside the tree-edit-distance scatter (A) and the bag/untrained-encoder divergence on identical forms (C), three views of the same leakage. It must be read with its non-learned rows: a bag over variable tokens alone scores 1.000 in *both* arms, and the trained encoder is below that in both. Leakage is heterogeneous across equations (ℓ from -0.05 to 1.12), so seed-level intervals understate equation-level uncertainty by $\sim 6\times$; we bootstrap over equations throughout.

Table 21: **The matched protocol-sensitivity contrast, with its skylines** (tags `r63`, `r69`). One equation set, one encoder trained once per seed and evaluated under both rows, one run family, the same three held-out forms per equation, and every equation's held-out slot drawn from the same construction at once. Training duplicates excluded; $K{=}8$, chance 0.125, $n{=}25$ seeds. Intervals are equation-level 95% bootstraps (Appendix E). **The non-learned rows are the context**: a bag over variable tokens alone identifies every held-out form in *both* conditions (S1 solvable by construction). The trained encoder is below that baseline in both arms.

Held-out form is…	Plain [95% CI]	Renamed [95% CI]
Trained Tree-LSTM:		
… a library paraphrase (in-library)	$0.965\ [0.950, 0.977]$	$0.880\ [0.842, 0.918]$
… an out-of-library equivalent	$0.832\ [0.727, 0.933]$	$0.662\ [0.483, 0.820]$
Structure-free, no training, same protocol:		
variable-token bag, in-library	$\mathbf{1.000}$	—
variable-token bag, out-of-library	$\mathbf{1.000}$	—
full token bag (in-lib / OOL)	0.980 / 1.000	—
variable-blind bag (in-lib / OOL)	0.715 / 0.600	—
untrained encoder (in-lib / OOL)	0.790 / 0.435	—

X. The AI Feynman Held-Out-Form Suite, Full Rows

Table 22: **The AI Feynman held-out-form suite, with both non-learned baselines** (n=25 seeds, 10 equations, chance 0.100). Encoder rows from tags `r21`/`r24`; bag rows from the E3 run. **The variable-only row is deterministic** (sd 0.000) rather than an estimate, and exceeds the trained encoder at every seed, so plain accuracy here carries no evidence about structural learning. Under within-range renaming the variable-blind bag still reaches ≈6× chance from operator content alone (S2). Full-token bag accuracy varies across runs with the held-out paraphrase draw: a replication (tag `r69`) gives 0.932 and random 0.748, straddling the trained encoder (Appendix D).

Model	Plain	Renamed (in-range)
Bag-of-tokens, variable-only (0 params)	$\mathbf{1.000 \pm 0.000}$	—
Bag-of-tokens, variable-aware (0 params)	0.992 ± 0.027	—
Trained Tree-LSTM	0.972 ± 0.060	0.824 ± 0.086
Random (untrained) encoder	0.724 ± 0.184	0.624 ± 0.145
Bag-of-tokens, variable-blind (0 params)	0.636 ± 0.111	—
Chance	0.100	0.100

Y. The Targeted Corpus Audit, Full Table

Table 23: **Zero-parameter separability survey across published expression benchmarks** (no training; tag `r57`, produced by the released auditor). *Unique var-set* (tier 1): fraction of equivalence classes whose variable *set* is shared by no other class. If high, knowing which variables appear pins the class, and a held-out-form protocol is solvable by token lookup. *Op-bag sep.* (tier 2): fraction of class pairs separable by the operator/arity bag alone. The tier-1 failure is confined to the physics block: the AI Feynman corpus, from which such benchmarks are built, gives 84% of equations a variable set no other equation shares. *The canonical published benchmark family for this task, EQNET (Allamanis et al., 2017), is immune*, defining every class over a shared variable pool ($\leq 3.2\%$ unique var-sets across all 15 corpora), as are generated symbolic-regression corpora. Tier 2, by contrast, is high nearly everywhere, which is why class-recovery claims must be stated relative to the operator-bag control. *These columns measure whether class representatives differ, not what a bag scores on held-out forms*; the two diverge sharply (Appendix S, Appendix P), and the released auditor prints the measured accuracy alongside them. N is the corpus's class count; corpora above 600 classes are subsampled (seeded) for the $O(n^2)$ pairwise columns. **The last two blocks port the auditor out of symbolic mathematics**, and only the second is a benchmark we did not build: identifiers for variables and AST node types with arities for the operator bag over 300 Type-2 clone classes α-renamed from *real* Python stdlib functions, *constructed*, so evidence the auditor ports and not of prevalence there, and T1 fires harder than in any symbolic corpus, 96% against AI Feynman's 84% (Appendix AW), and **SCAN** (Lake & Baroni, 2018), *published*, whose classes are its own gold action sequences, with content words for variables and its connectives for the operator bag. **SCAN is immune at T1 like EQNET, yet a content bag still scores $64\times$ chance on held-out forms**: the sharpest instance in this table of the representatives-versus-held-out divergence noted above (Appendix AX).

Benchmark	N	Var. pool	Unique var-set (T1)	Op-bag sep. (T2)
Physics-derived held-out-form benchmarks (one variable subset per equation):				
AI Feynman (Udrescu & Tegmark, 2020)	100	93	**84**%	99%
Feynman-10 (this work)	10	15	**100**%	98%
Synthetic 8-law (this work)	8	11	**100**%	96%
EQNET / semvec (Allamanis et al., 2017), the published equivalence-class family:				
POLY (7 corpora)	47–1102	1–3	0%	66–92%
BOOL (8 corpora)	95–7312	3–10	0–3%	69–97%
Generated SR / integration corpora (shared variable pool):				
NeSymReS (Biggio et al., 2021)	200	3	0%	100%
Lample–Charton (Lample & Charton, 2020)	320	1	0%	99%
Shared-variable designs (this work, the fix):				
Feynman-10 shared (de Bruijn)	10	4	20%	98%
Overlap-5 (shared x_0, x_1)	5	2	0%	100%
Overlap-12 (shared x_0, x_1)	12	2	0%	100%
Outside symbolic mathematics: code, a corpus we constructed (tag `r95`):				
Python stdlib Type-2 clones	300	1454	**96**%	**100**%
*Outside symbolic mathematics: a **published** benchmark (tag `r96`):*				
SCAN (Lake & Baroni, 2018)	9222	7	0.3%	99%

Z. The UnseenEqClass k-NN Protocol, Fully Specified

This appendix specifies the OOD protocol of Table 27 completely, with a toy example and a no-leakage argument, because the class-precision-on-held-out-classes design is the one a reviewer most needs to be able to reconstruct and check for leakage (tag `r70`).

Construction. Load the poly8 corpus (1102 equivalence classes over the shared pool $\{x_0, x_1, x_2\}$), keep the $|C| \geq 2$ classes, and fix a single shuffle (seed 70). Split the shuffled class list by *index* into two disjoint blocks:

$$\text{TRAIN} = \text{usable}[0\!:\!K], \qquad \text{UNSEEN} = \text{usable}[K : K + \min(\lfloor K/5 \rfloor, 100)].$$

The encoder is trained (triplet + BCE) using *only* the forms of TRAIN classes. No form, label, centroid, or hyperparameter is ever computed from UNSEEN.

The two metrics.

- SEENEQCLASS (in-distribution): for each TRAIN class, one form is held out; it is scored correct if its nearest *training-class centroid* (mean of that class's remaining forms) is its own. Chance $= 1/K$.
- UNSEENEQCLASS (OOD): embed *all* forms of the UNSEEN classes. For each such form q, take its k=5 nearest neighbours *among the other UNSEEN forms* (query self-excluded); class precision is the fraction of those 5 that carry q's class label, averaged over all UNSEEN query forms. The neighbour pool is UNSEEN itself: the metric never consults a training centroid, so it measures whether the frozen encoder maps forms of a *never-seen* class near one another. Its null and its attainable maximum are both derived below; **neither is** $1/|\textbf{UNSEEN}|$ **nor** $1/(\textbf{number of unseen classes})$, which is what earlier versions of this paper reported.

The null, corrected (tag `r80`). Because SEENEQCLASS is 1-of-K identification, $1/K$ is exactly its null and is unaffected by anything here. UNSEENEQCLASS is not an identification task, and its null is governed by class *sizes* rather than class *count*. Let the UNSEEN pool hold N forms in classes of sizes $\{m_c\}$. Under random class labels, one neighbour of a query in a class of size m is a class-mate with probability exactly $(m{-}1)/(N{-}1)$, both -1s come from excluding the query itself. This is independent of k, so the null is *flat* in k even though the metric is not. Averaging over queries in the same query-weighted way the metric does,

$$\mathbb{E}[\text{precision}] \;=\; \frac{\sum_c m_c\,(m_c-1)}{N\,(N-1)}.$$

This equals $1/(\text{classes})$ only for equal-sized classes and large N. Poly8 is loaded at `max_forms_per_class`=20 and its sizes are heavy and uneven (at K=500: 100 classes, N=731, sizes min 2, median 6, mean 7.3, max 20, ten classes at the cap), so the value we published was too *small*: $0.100/0.050/0.025/0.010/0.012$ against the exact $0.152/0.082/0.035/0.014/0.016$ at K=$50/100/200/500/1000$, an understatement of $1.52/1.63/1.38/1.37/1.31\times$. **The correction therefore runs against every claim we make**, shrinking each reported margin; it changes no ordering, because every row is measured under one protocol and only the reference line moves. We verify it two independent ways: 2000 label permutations scored through each representation's *own* neighbour graph (holding the tie structure fixed, which matters for bags that map distinct forms to identical vectors by design), and 2000 permutations through a uniformly random graph, which shares no code with the closed form. Both agree with it to $\leq 1.3\times10^{-3}$ at every K and every k, and every row of Table 26 clears the 97.5th percentile of its own permutation null.

The attainable maximum. The same class-size distribution bounds the metric from above. A query whose class holds m forms has only $m{-}1$ class-mates in the pool, so at most $\min(m{-}1, k)$ of its k neighbours can share its label, and the best score *any* representation can achieve is

$$\max[\text{precision}] \;=\; \frac{1}{N}\sum_c m_c\,\frac{\min(m_c-1,\,k)}{k}.$$

At k=5 this is $0.845/0.905/0.909/0.908/0.876$ across the five scales, never 1.000, because the minimum class size is 2. This reinterprets a caveat we raised ourselves. UNSEENEQCLASS appeared strongly k-dependent (Tree-LSTM at K=500: 1.000, 0.965, 0.894, 0.710 at k=1, 3, 5, 10), which we had presented as a limitation of the result; but the attainable maximum falls in near-lockstep (1.000, 0.969, 0.908, 0.723), and expressed as a fraction of what is attainable the encoder holds 0.96–1.00 at every k and every K. Almost all of the apparent decay is a property of the metric, not of the representation. Two things follow: the choice k=5 needs no defence beyond this table, and the headline is more precisely stated as *near-saturation of a metric whose maximum is below one* than as 0.894 out of 1.000. Both the null and the attainable maximum are functions of the label vector alone, which the seed-70 split fixes, so both are computed without retraining and apply unchanged to the trained rows.

Toy example. Take K=10 and UNSEEN $= \{A, B\}$ with forms A=$\{a_1, a_2, a_3\}$, B=$\{b_1, b_2\}$, so N=5 (the code requires $\lfloor K/5 \rfloor \geq 1$, which rules out a degenerate split). We embed all five forms with the encoder trained on the ten TRAIN classes. For query a_1 at k=2, the neighbours are drawn from $\{a_2, a_3, b_1, b_2\}$; if they are a_2, a_3 the precision for a_1 is 1. Averaging over all five queries gives the cell. The three quantities of this appendix, at k=2: the *null* is $[3{\cdot}2 + 2{\cdot}1]/[5{\cdot}4] = 0.40$, not $1/2$; the *attainable maximum* is $[3{\cdot}\frac{2}{2} + 2{\cdot}\frac{1}{2}]/5 = 0.80$, not 1, because b_1 and b_2 have only each other; and

a perfect encoder scores that 0.80. Because A and B were absent from training, a score above the null can only come from the encoder placing equivalent unseen forms together.

No-leakage argument. TRAIN and UNSEEN are disjoint by index into one fixed shuffle, so no class appears in both. Three consequences: (i) *no label leakage*, since UNSEEN class identities are never a training target; (ii) *no representation leakage through selection*, since the shuffle seed and split point are fixed *before* training and independent of any embedding; (iii) *no tuning leakage*, since all hyperparameters (margin, epochs, k) are constants, never selected on UNSEEN. The only channel from training to the UNSEEN score is the frozen encoder's weights, which is exactly the generalization we intend to measure. Poly8 is additionally S1 immune (shared pool) and S2 immune (0/1102 unique operator bags), so a query form cannot be matched to its class by variable identity or operator multiset; and the non-learned baselines of Table 27 are run through the *identical* split and metric, so the trained–baseline comparison is like-for-like.

AA. The Probe Ladder: Operand Relocation, and How Much of S3 the Twin Test Spans

This appendix does two different jobs with one matrix, and they are billed differently.

The first job, and the one it was run for. The shape-matched swap of §4.2 still changes left/right order, so it could be read as testing argument order alone. Operand relocation removes that reading. We do *not* treat it as part of the central argument: exchanging two leaves holding *different* variables leaves only the variable-to-position binding, and variable-to-position binding is a structural relation tied closely to variable identity, so it is not cleanly axis C under our own definition (§3.2). What the control establishes is narrower and still worth having: *the residual is not explained solely by the tested argument-order statistics*.

The second job, which is what §3.2 cites, and which the caveat above does not touch. Admitting a control by a twin test is a *membership* test, never a proof of global invariance (AB): a candidate can be blind to the twin we built and sensitive to a perturbation we did not. That is the honest limitation of admissibility auditing, and the objection it invites is *how large the unspanned residual is*. **Because the three probes here are *nested*, each matches everything the previous one matched and one thing more, the same matrix answers that question, without being asked to support any claim about composition.** Table 24 reads it that way. Two things it shows. **One rung looser than the twin the body reports, the test admits a control that scores** 0.909 **on the very perturbation membership requires it not to see**, so the twin's matching is doing measurable work, and a weaker test would have passed a control that a stronger test rejects. **And at the strictly most-matched rung, where all three bags are pinned, the encoder gap stays positive**, so the verdict the body reads off the middle rung is not an artefact of stopping there.

Table 24: **Takeaway: the twin test's coverage is measured, not assumed**, three *nested* probes (tag `r74`), each matching everything the one above it matches and one property more. **"Bags pinned" counts how many of the same three non-learned bags the probe pins at exactly** 0.500 **by construction**, and **"strongest unpinned" is the highest score any *unpinned* bag reaches on that probe**, the control a test at that rung would admit while it is demonstrably sensitive to the perturbation. Encoder-gap ranges are over the cells carrying an encoder row; BOOLEAN8 carries none in this matrix (its three cells are construction-checked only) and ONEVARPOLY13 admits no relocation, which is why the cell counts differ by rung.

Probe	Holds invariant	Cells	Bags pinned	Strongest unpinned	Trained−untrained gap (cells with an encoder)	
`rotate`	inventory	6	$1/3$	tree-local $\mathbf{0.909}$	$+0.060$ to $+0.414$	(5)
`swap`†	+ local shape	6	$2/3$	token n-gram 0.769	$+0.103$ to $+0.252$	(5)
`relocate`	+ argument order	5	$3/3$	*none left unpinned*	$+0.142$ to $+0.383$	(4)

† The rung the body reports as *the* shape-matched twin (§4.2). All 17 cells pass their construction check, and the gap is positive in all 14 cells that carry an encoder row.

Construction (tag `r74`). *Operand relocation* exchanges two leaves holding different variables, $(x_0 - x_1) \cdot x_2 \to (x_0 - x_2) \cdot x_1$. The S3 baggers being variable-blind, inventory, local shape *and* argument order are all invariant under this transformation, verified per instance.

Result. Across 17 cells over six corpora and two symbolic families every construction guarantee held, and the encoder gap was positive in all 14 of those cells that carry an encoder row. Under relocation the trained–untrained gap is $+0.142$, $+0.383$, $+0.210$, $+0.184$, the same range as the swap twin, so the swap-twin residual is not attributable to argument order. The matrix records its own limits: ONEVARPOLY13 admits no relocation (it has one variable), and BOOLEAN8 carried no encoder row *in this matrix*, the encoder vocabulary at the time of that run holding no boolean operators, a gap since closed: Appendix AO extends the vocabulary and runs the swap twin on that corpus with all three architectures.

AB. STRONGER NON-LEARNED BASELINES ON THE SHAPE-MATCHED TWIN, FULL LADDER

Table 25 gives the full ladder summarised in §4.2: the non-learned baselines a skeptic would reach for, run on the identical 348 swap-twin classes, plus the rotation-twin contrast. The trained–untrained gap is reported as a class-level interval rather than a spread over initializations, and because one class *is* one twin the two arms are paired: re-running both with per-class capture (`r104_swap_twin_classci`, the same three seeds, reproducing 0.9943 and 0.7883 exactly) and bootstrapping gap_i over all 348 classes gives 0.206 $[0.172, 0.240]$, excluding zero. That gap is to the untrained encoder, which is *not* a member of $\mathcal{F}_3$; the criterion gap of Definition 1 is to the 0.500 the family is pinned at.

What the 0.500 pinning does and does not establish. A baseline pinned at exactly 0.500 on every one of the 348 classes is admitted to $\mathcal{F}_3$ *under this operational definition*: it cannot see the swap these twins make. **That is a membership test, not a proof of global invariance to S3**, the same baseline could be sensitive to some other rearrangement we did not construct a twin for, and the rotation-twin column is a case where the ordering of the very same baselines changes. Admissibility is therefore relative to the declared family and to the published test, exactly as passing the $\mathcal{F}$-relative admissibility audit is relative to that family (Definition 1); the test is released so a reader can admit a candidate we never considered, or exhibit a twin that rejects one we did.

Table 25: **Stronger non-learned baselines on the shape-matched twin** (poly8, K=500, 348 classes, chance 0.5; tags `r75_stronger_baselines` for the non-learned ladder, `r73_swap_twin` for the encoder rows, `r72_structure_twin` for the rotation contrast). Every order-blind member of the ϕ_d family (WL kernel, bounded-depth histogram) is pinned at 0.500 by construction; tree-edit distance (0.723) is the only non-learned baseline that separates a sibling swap, and it still trails the untrained encoder. The untrained encoder needs recursion depth ≥ 4 to reach 0.788; the remaining 0.206 to trained 0.994 is closed only by learned parameters. Rotation-twin values (right) are shown for contrast: there the same baselines *exceed* the tree-local bag, which is why rotation is the weaker probe. Class-level 95% bootstrap CIs in brackets; unbracketed cells are pinned at 0.500 by construction or are encoder seed means that no per-class rerun covers.

	Swap twin (shape-matched)	Rotation twin
Full-token bag	0.500	0.500
Tree-local bag (S3, ϕ_1)	0.500	0.861 $[0.828, 0.892]$
WL subtree kernel (h=1, 2, 3)	0.500	0.889 $[0.858, 0.916]$
Bounded-depth ϕ_d (d=1)	0.500	0.889 $[0.858, 0.916]$
Bounded-depth ϕ_d (d=2, 4, 8)	0.500	0.901 $[0.873, 0.928]$
Token n-gram (order)	0.681 $[0.647, 0.716]$	0.559 $[0.511, 0.608]$
Tree-edit-distance NN (order)	0.723 $[0.693, 0.753]$	0.889 $[0.862, 0.914]$
Untrained encoder, depth 0–1	0.500	0.767
Untrained encoder, depth 2	0.562	0.748
Untrained encoder, depth ≥ 4	0.788 $[0.754, 0.823]$	0.943
Trained Tree-LSTM	$\mathbf{0.994}$ $[0.986, 1.000]$	$\mathbf{1.000}$

AC. Where Known Structural Baselines Sit Inside the ϕ_d Family

§3.2 defines the bounded feature family $\phi_d(T)$ and states that a S3 result rules out sensitivity to ϕ_d only for the d actually tested. This appendix places the standard structural baselines inside that family, which is what makes the sweep in Table 27 a sweep over the family rather than over three unrelated methods.

- **Our `tree_local` bagger** is essentially the $d{=}1$ member: parent–child operator pairs plus a subtree-size histogram, both readable from depth-1 truncations.
- **The Weisfeiler–Lehman subtree kernel** (Shervashidze et al., 2011) to height h is the rounds-based analogue of ϕ_d: each WL round propagates a hashed label one hop, so height h reads a neighbourhood of radius h exactly as ϕ_d reads a truncation of depth d. Both are order-blind on unordered children, which is why both are pinned at 0.500 on the sibling-swap twin.
- **Tree-edit distance** is the order-*sensitive* extension: it is not a member of the order-blind family at all, and it is the only non-learned baseline in this paper that can separate a sibling swap (0.723, Appendix AB). Its inclusion is why "exceeds every non-learned baseline" is not satisfied merely by order-blindness.
- **The limit, measured rather than argued.** On poly8 the family *saturates*: the corpus's maximum tree depth is 4, so $\phi_{d\geq 4}$ already emits the complete order-blind fingerprint; we verify this as *bag equality* between $\phi_{d=8}$ and the uncapped order-blind bagger at all five K, not as a bound. Its order-sensitive completion ϕ_∞ (each node's entire subtree, children in order) determines T up to variable renaming, and it scores *below* the coarse $d{=}1$ bag at every scale (Table 27) while being the *less* tied of the two for $K{\geq}100$, so the loss is discrimination misdirected, not discrimination lost: a held-out form's root string matches no other form's, and a fingerprint that separates everything groups nothing. Resolution in d is therefore **non-monotone**, which is what forbids both rescuing the family with a larger d and dismissing the sweep for using a small one. S3 and composition are separated by boundedness rather than categorically, and this is the precise reason clearing the criterion cannot certify composition.

AD. EQNET poly8: Both Protocols, All Architectures

Table 26 is the OOD UnseenEqClass ladder §4.2 quotes from; every value the main text states is in it. Table 27 then adds the in-distribution SeenEqClass block,

where GIN and Transformer degrade sharply as K grows while retaining strong unseen precision, a split we note but do not explain.

AE. Auditing the Shared-Variable Positive Result at S2

§4.1 reports that the shared-variable Feynman redesign is the one physics-suite result clearing every non-learned baseline available to it. We audited it by the same standard we apply to published work.

Operator scrambling (tags `r53`, `r54`, `r61`). Each held-out operator is replaced by a random one of the same arity, leaving tree shape and variable content intact. Trained identification falls from 0.932 to 0.312 $[0.184, 0.460]$ while the random-encoder skyline falls to 0.140, a trained-minus-random gap of $+0.172$ $[0.044, 0.300]$ (bootstrap over equations). The collapse locates the dominant cue in operator content, and the surviving gap is the shape residual.

Arity control. Restricting to the six two-variable equations, so arity cannot disambiguate, identification is 0.887 $[0.753, 1.000]$ against within-group chance 0.167. The result survives arity control but rests on S2 information, which is why we state it as identification *above the operator-bag skyline* rather than as structural identification.

AF. The AI Feynman Case Study, Full Controls

§4.1 states the AI Feynman result in one paragraph because it is a case study of the audit, not a finding about the field. The controls behind it are here.

Table 26: **EQNET `poly8`, UNSEENEQCLASS**: k-NN class precision (k=5) on classes *never seen in training*, the OOD test (5 seeds; brackets are 95% bootstrap intervals over *equivalence classes*, the unit of inference). The null is the *exact* random-label expectation $\sum_c m_c(m_c-1)/N(N-1)$ over unseen class sizes, matched to $\leq 1.3\times10^{-3}$ by a 2000-draw permutation; the attainable maximum is the best *any* representation could reach, a query's class supplying at most m_c-1 of its k neighbours; ϕ_∞ is the family's order-sensitive *completion*, each node's *entire* subtree, children in order. Architectures, further baselines, both protocols, the k-sweep and all derivations: Appendices Z, AD, AI. **A correction of our own labelling, made in round 36 and running *against* us**: earlier drafts printed the tree-local bag as "ϕ_1" and read $\sup \mathcal{F}_3$ off it. It is a *distinct* descriptor of ours, depth-indexed labels, parent–child pairs and subtree sizes in one bag, the statistics §3.2's S3 row names, and Definition 1 does not enumerate it, so it is reported but *excluded* from the supremum. It exceeds $\sup \mathcal{F}_3$ at three of the five scales, which is why the correction *lowers* the bar we quote ($0.277 \to 0.276$ at K=500) rather than raising it; no verdict moves, because the strongest non-learned baseline here is the full-token bag either way. Tags `r70_poly8_structural_v2`, `r71_tier3_baselines_v3`, `r80_null_calibration`.

	K=50	K=100	K=200	K=500	K=1000
Tree-LSTM	**0.843**	**0.882**	**0.872**	**0.894**	**0.868**
95% CI over classes	[.61, .95]	[.76, .94]	[.80, .92]	[.86, .92]	[.82, .91]
Random (untrained) encoder	0.711	0.673	0.673	0.525	0.538
Bag (full token), strongest non-learned	0.794	0.770	0.673	0.517	0.509
95% CI over classes	[.57, .90]	[.60, .88]	[.56, .77]	[.43, .59]	[.42, .60]
Bag ($\sup \mathcal{F}_3$, the audit statistic)	0.736	0.641	0.488	0.276	0.278
Bag (tree-local, S3; $\notin \mathcal{F}_3$)	0.726	0.649	0.493	0.277	0.277
Bag (ϕ_∞, *complete* fingerprint)	0.490	0.549	0.330	0.237	0.199
Attainable max (k=5, any representation)	0.845	0.905	0.909	0.908	0.876
Random-label null (exact)	0.152	0.082	0.035	0.014	0.016

The token-bag skyline. Under plain held-out evaluation the Tree-LSTM reaches 0.972 ± 0.060 (n=25 seeds, chance 0.10) against a random encoder's 0.724 ± 0.184. Because every equation in the 10-equation suite has a variable set no other equation shares (Table 23), a variable-token bag is a complete lookup solution and scores 1.000 (sd 0.000), above the trained encoder at every seed. Table 22 (Appendix X) gives the full rows, including a replication in which the *full*-token bag falls below the trained encoder, which is why the comparison is anchored on the deterministic variable-only row rather than a near-tied one.

Renaming and coverage controls. Under within-range renaming the (plain, renamed) pair is $(0.972, 0.824)$ for the trained encoder and $(0.724, 0.624)$ for the random one, while the variable-blind bag scores 0.636: variable identity is the dominant cue and S2 is what remains. With genuine out-of-library rewrites the pair collapses to $(0.50, 0.15)$ (Appendix E), which is the coverage effect of §4.5 showing up on this suite.

Why this is a counterexample and not a prevalence estimate. The suite is 10 equations selected for 1–3 variables and closed-form structure, under a per-symbol tokenization convention. It demonstrates that the failure mode is *realisable*, and the 23-corpus audit (§4.1) is what estimates how common it is. The published equivalence-class family is immune at S1 by design.

AG. AUDITING EQNET UNDER ITS OWN SPLITS

Variable identity is semantic in the EQNET corpora, so they are S1 immune by construction. Running the full diagnostic under *their* splits (tag `r66`): on SIMPLEPOLY8 (86 classes, chance 0.012) the trained encoder reaches 1.000 [1.000, 1.000] against a bag at 0.242 and a random encoder at 0.115, with zero of 747 held-out forms string-identical to a training form; on their UNSEENEQCLASS split it reaches 0.980 against a bag at 0.741. On the smaller SIMPLEPOLY5 unseen split the bag nearly matches the encoder (0.433 vs. 0.467), a corpus where the checklist's stage 2 would have stopped a structural claim before any model was trained.

Table 27: **EQNET `poly8`, both protocols** (Tree-LSTM rows tag `r70_poly8_structural_v2`, GIN and Transformer tags `r70_poly8_gin_tf_v2` and `r70_poly8_gin_tf_v2_lo`, non-learned rows tag `r71_tier3_baselines_v3`; 5 seeds, 10 epochs). SEENEQCLASS is centroid identification of the held-out form of a *training* class, so $1/K$ is exactly its null; UNSEENEQCLASS is k-NN class precision on classes never seen in training, whose null and attainable maximum are both set by class *sizes* and are derived in Appendix Z, the superseded $1/(\text{unseen classes})$ row is retained so the correction is visible rather than overwritten. The unit of inference is the equivalence class, not the seed; class-level 95% bootstrap CIs for the Tree-LSTM seen row in brackets. ϕ_∞ is the ϕ_d family's order-sensitive *completion* (each node's entire subtree, uncapped, children ordered): complete up to variable renaming, and *weaker* than the coarse tree-local bag on both protocols at every scale. †the only cells not re-run under the corrected tie rule: GIN and Transformer at $K{=}1000$ are read from tag `r70_poly8_k1000`, which predates it; the re-run at the other four scales reproduced all 16 carried-over cells to within 0.030 (median 0.005), so no claim changes. **That 0.030 is irreproducibility of the same computation, not seed variance**: GIN and Transformer training in this harness is not run-to-run reproducible on MPS at all, while Tree-LSTM training is bit-for-bit (Appendix AJ). Every GIN and Transformer cell here therefore carries a ± 0.03 run-to-run uncertainty that no interval printed expresses; the Tree-LSTM and non-learned rows do not (`REVISION_HISTORY.md` §9).

	$K{=}50$	$K{=}100$	$K{=}200$	$K{=}500$	$K{=}1000$
SeenEqClass (centroid identification, in-distribution):					
Tree-LSTM	0.976	0.984	0.992	0.990	0.980
95% CI over classes	[0.94, 1.00]	[0.97, 1.00]	[0.98, 1.00]	[0.98, 1.00]	[0.97, 0.99]
GIN	0.968	0.922	0.794	0.680	0.527†
Transformer	0.728	0.700	0.662	0.555	0.443†
Random encoder	0.628	0.642	0.587	0.476	0.436
Bag (full token)	0.680	0.500	0.390	0.212	0.147
Bag (ϕ_∞, *complete* fingerprint)	0.220	0.160	0.095	0.040	0.021
Bag (variable-blind)	0.200	0.130	0.065	0.024	0.012
Chance $= 1/K$ (identification; *unchanged*)	0.020	0.010	0.005	0.002	0.001
UnseenEqClass (k-NN precision, OOD classes never seen in training):					
Tree-LSTM	**0.843**	**0.882**	**0.872**	**0.894**	**0.868**
GIN	0.835	0.874	0.857	0.745	0.763†
Transformer	0.823	0.817	0.836	0.728	0.736†
Random encoder	0.711	0.673	0.673	0.525	0.538
Bag (full token)	0.794	0.770	0.673	0.517	0.509
Bag (tree-local, S3; $\notin \mathcal{F}_3$)	0.726	0.649	0.493	0.277	0.277
Bag (WL/ϕ_d family, best) $= \sup \mathcal{F}_3$	0.736	0.641	0.488	0.276	0.278
Bag (token n-gram, S3)	0.426	0.603	0.307	0.230	0.209
Bag (ϕ_∞, *complete* fingerprint)	0.490	0.549	0.330	0.237	0.199
Bag (variable-blind)	0.561	0.516	0.231	0.146	0.115
Attainable max ($k{=}5$, any representation)	0.845	0.905	0.909	0.908	0.876
Random-label null (exact)	0.152	0.082	0.035	0.014	0.016
superseded $1/(\text{unseen classes})$	0.100	0.050	0.025	0.010	0.012

AH. COMPOSITION OVER THE REWRITE LIBRARY (TAGS `R76`, `R79`; `R81`'S FULL MATRIX)

Superseded, and kept only for continuity. Appendix AU runs this design to depth 8 with three held-out conditions and is what the main text cites; nothing here is needed to follow the paper. It is retained because its numbers were published in an earlier version and because the depth-8 run inherits its construction and closure guard unchanged, **a reader checking the current claims can skip to AU**. **One table here is current and is not superseded**: the complete training-depth × test-depth matrix below (Table 29, tag `r81`), which §4.3 quotes two rows of in prose and which was placed here for space, not because it was replaced.

Every other probe in this paper tests sensitivity to a distinction between two forms that are *not* algebraically equivalent. That is a real limitation: an encoder holding ordered syntactic representations passes such a probe without knowing any algebra. This appendix documents the experiment that objection implies, and it is the only family of result we state as *systematic generalization to unseen compositions of learned rewrite operations* rather than as structural discrimination.

Construction. We run the experiment on *two* pairs of semantics-preserving schemas drawn from the 13-element library, because the two measure different things and a ceiling on the easier one cannot answer the harder question. Applicability was measured before any accuracy was looked at: only four schemas apply to essentially every `poly8` anchor (`add_identity` 120/120, `mul_identity` 120/120, `double_negate` 120/120, `commute` 113/120), which is what makes these the two feasible pairs; `distribute` (25/120), `power_to_explog` (40/120) and `reassoc` (0/120) apply too rarely to support a matched two-arm design.

- IDENTITY pair, $A:\ a \to a \cdot 1$, $B:\ a \to a + 0$. The *cleanest* composition available: both apply everywhere, so no class is dropped. Also the easiest, since each insertion only pads the tree.
- INVASIVE pair, A swaps the two arguments of a $+$ or $\cdot$ node (changing argument order, adding no nodes), $B:\ x \to 0 - (0 - x)$ (adding four nodes and two levels of depth). This is the harder question: the composite differs from its anchor in order *and* in shape.

A composite $A \circ B$ form is *genuinely equivalent* to its anchor in both pairs. Two arms are matched on training-form count so that library *membership* is the only difference:

- PRIMITIVES: the $k{=}4$ training paraphrases are *single* applications of A or of B. No composite is ever seen.
- COMPOSITES: the same count, with two of the four replaced by $A \circ B$ composites applied at sites other than the test form's.

The held-out form is an $A \circ B$ composite, byte-identical across both arms and excluded from the composite pool so that even the COMPOSITES arm never trains on that exact tree. $K{=}200$ classes, 5 seeds, chance 0.005.

Construction guarantee, recomputed not asserted. If a held-out composite were reachable from its anchor by a *single* rewrite, the PRIMITIVES arm could have reached it without ever composing and the result would be void. The check is therefore membership, not arithmetic: over all classes `verify_claims.py` enumerates the single-application closure of each anchor under A and under B at every eligible site and recomputes that the held-out form lies in neither, and that no held-out form appears in either arm's training pool. A node-count argument would have been pair-specific and silently wrong: it holds for the IDENTITY pair (each insertion adds exactly two nodes, so a primitive is $|\text{anchor}|{+}2$ and a composite $|\text{anchor}|{+}4$, also recomputed, for that pair only), but `commute` adds no nodes at all, which would leave an invasive primitive and its composite indistinguishable by size.

Result. Class-bootstrap means over 200 classes, 5 seeds, chance 0.005:

Arm	IDENTITY pair	INVASIVE pair
PRIMITIVES (composition never seen)	$1.000\ [1.000, 1.000]$	$0.993\ [0.987, 0.998]$
COMPOSITES (composition in library)	$0.999\ [0.997, 1.000]$	$0.997\ [0.992, 1.000]$
COMPOSITES − PRIMITIVES	$-0.001\ [-0.003, 0.000]$	$+0.004\ [0.000, 0.009]$
Untrained encoder (same architecture)	$0.409\ [0.375, 0.444]$	$0.419\ [0.384, 0.454]$
Strongest non-learned baseline	$0.415\ [0.345, 0.485]$	$0.135\ [0.090, 0.185]$

Composing is close to free in both pairs: the arm that never saw a composite matches the arm that trained on composites, and the difference interval excludes any deficit larger than 0.003 (IDENTITY) or 0.009 (INVASIVE). Neither ceiling is an artifact of an easy task, and the INVASIVE pair is the sharper evidence: the identical held-out forms leave an untrained encoder at 0.419 and *every* non-learned baseline at or below 0.135, so the trained encoder clears the strongest one by more than 0.85 absolute where on the IDENTITY pair it clears it by 0.585. Reordering arguments and changing depth is what breaks the bags, and it is exactly what the encoder is unaffected by.

Non-learned ladder on the identical held-out composites. The full `BAGGERS` ladder, run under the same centroid protocol on the same forms, for both pairs. Every value is re-scored with centroid ties broken label-blind (tag `r78`, `--r76_bags`; the rule and why it was needed here too are in Appendix AI), so these supersede the `argmax` values of the run log:

Baseline	IDENTITY	INVASIVE
Bag (full token)	0.415	0.125
Bag (token n-gram)	0.085	0.135
Bag (tree-local, S3; $\notin \mathcal{F}_3$)	0.105	0.110
Bag (WL h=1 / 2 / 3; S3)	0.055 / 0.070 / 0.070	0.060 / 0.075 / 0.085
Bag ($\phi_{d=1}$ / 2 / 4 / 8; S3)	0.045 / 0.045 / 0.030 / 0.030	0.080 / 0.060 / 0.060 / 0.060
Bag (variable-only)	0.100	0.105
Bag (variable-blind)	0.075	0.025
Chance	0.005	0.005

In neither pair is the composite identifiable from inventory or from any tested member of the bounded ϕ_d family. The two columns differ instructively: on the IDENTITY pair the full-token bag reaches 0.415, because padding a tree with $\cdot 1$ and $+0$ leaves most of the token multiset intact and adds a distinctive constant, whereas on the INVASIVE pair it collapses to 0.125 and *every* baseline sits in $[0.025, 0.135]$. The S3 requirement of §3.2 holds against the family in both, and holds with far more room in the pair that actually rearranges the tree.

Why this is the stronger probe. The twin probes ask the encoder to separate a form from a non-equivalent look-alike, which ordered syntax suffices for. Here the held-out form *is* algebraically equivalent to its anchor, it was produced by composing two rewrites the encoder only ever saw applied singly, and no bounded local statistic recovers it. What remains to be explained is the composition itself.

What this does not show, and what `r79` has since shown. Both pairs here are depth-2 compositions of two schemas, and all four arms sit above 0.99; on its own that establishes only that composition of *these* primitives is close to free, not that composition is free in general. An earlier version of this appendix went on to say that establishing depth-scaling would need a corpus built for it, a benchmark-design contribution rather than an audit. That was wrong, and cheaply so. Extending the INVASIVE chain to A, B, A, B costs no coverage at all (only `commute` is conditional, and it fails on the same 14 single-variable anchors at every depth) so the comparison is *paired* over one fixed class set, and Table 28 below reports depths 1–4 plus the reversed order. It also locates a boundary this appendix could not see: the cost of never having composed is $+0.017$ at d=2 but $+0.037$ at d=4. We leave the retracted sentence visible rather than deleting it, but do *not* add it to the four-claim tally of §1, because what changed it was running a further experiment, not applying the audit. Its numbers also differ from this appendix's (0.967/0.984 at d=2 against 0.993/0.997 here), because `r79` restricts to the harder anchors admitting the full depth-4 chain; both stand, and the table caption says so. What remains out of reach is the two structurally richest schemas, and there the reason is a corpus limit rather than a choice: `distribute` applies to 25 of 120 anchors and `reassoc` to none, so a matched two-arm design over them would either drop most classes or silently change the corpus. We also note the asymmetry with §4.5: A and B are *in* the training library and only their composition is withheld, whereas the out-of-library arm withholds the schema itself and does not transfer. The two results are consistent, composition inside a covered library generalizes, degrading measurably with depth; extension beyond the library does not, and together they say that coverage, not compositional depth, is what the Feynman failure measures.

The two-schema fixed path, kept as a continuity check (tag `r79`). Table 28 is the curve §4.3 reported before the primitive set replaced it, and it is retained here for two reasons rather than deleted. First, continuity: `r76`'s depth-2 cells, this table's, and `r81`'s are three measurements of overlapping quantities on one corpus, and a reader should be able to see that they differ only where the anchor sample or the basis differs, not arbitrarily. Second, and less comfortably, it is the evidence for the sentence §4.3 now has to write: over this narrow basis the cost of never having composed is $+0.037$ at d=4, and over `r81`'s four primitives and 24 orders it is $+0.069$, so the earlier near-flat reading was in part a property of testing two schemas along one path. The design that produced it was sound and its numbers stand; what it could not do is distinguish composing from holding two invariances, which is exactly why the basis, not the depth, was the parameter to vary next.

The full four-primitive matrix, demoted from §4.3. §4.3 quotes in prose the two rows the argument turns on, the never-composed arm and the fully covered library, and the whole design is

Table 28: **The two-schema fixed path** (poly8, K=200, 5 seeds, centroid identification, chance 0.005; tag `r79`). Superseded in §4.3 by Table 29, which widens the basis from two schemas on one path to four primitives over 24 orders. Depth d holds out a chain of d rewrites along the single path A, B, A, B; the never-composed arm trains only on *single* applications and is evaluated at every d, so columns are paired over one class set. The d=3 chain is its own reverse. Depth 2 uses `r76`'s schema pair, which reported 0.993/0.997 over its own easier anchor sample; both stand. Non-learned rows are re-scored under the label-blind tie rule this run had not inherited (tag `r78_r79_bag_ties`): two cells move by 0.005, no comparison (Appendix AI).

	d=1	d=2	d=3	d=4
Trained, **composition never seen**	**0.965**	**0.967**	**0.961**	**0.945**
95% CI over classes	[.94, .98]	[.95, .98]	[.94, .98]	[.92, .97]
Trained, composites in library (upper bound)	—	0.984	0.987	0.982
cost of never composing, paired	—	+0.017	+0.026	**+0.037**
95% CI, paired over classes		[.007, .030]	[.013, .041]	[.024, .052]
Reversed chain, same encoder	—	0.986	*self-rev.*	0.950
Untrained encoder	0.427	0.248	0.278	0.153
Strongest non-learned (full-token bag)	0.330	0.190	0.190	0.115
S3 maximum (tree-local bag)	0.135	0.095	0.095	0.055
ϕ_∞, *complete* fingerprint	0.030	0.030	0.025	0.025

Table 29 below. The two middle training depths are what make the closure *monotone* rather than a two-point comparison, and the alternate-order row is what rules out the reading that one lucky permutation carries the result.

Table 29: **The complete training-depth × test-depth matrix** (poly8, K=200, 5 seeds, centroid identification, chance 0.005; tag `r81`). Rows marked $\mathcal{X}$ are structural stress tests outside $\mathcal{F}_3$. Every above-diagonal cell holds at ≥0.993, which is what licenses reading the deficit as one of *coverage* rather than of depth.

training depth ↓ / *test* depth →	d=1	d=2	d=3	d=4
1: composition never seen	0.997	0.983	0.950	0.918
95% CI over classes	[.99, 1.0]	[.97, .99]	[.94, .96]	[.90, .94]
2: depth-2 composites in library	0.998	0.994	0.982	0.955
3: depth-3 composites in library	0.997	0.994	0.985	0.980
4: depth-4 composites in library	0.997	0.996	0.993	0.987
***cost of never composing*, paired**	—	+0.011	+0.035	**+0.069**
Alternate order, same multiset	—	0.996	0.970	0.937
Untrained encoder ($\mathcal{X}$)	0.482	0.379	0.289	0.215
Strongest non-learned (full-token bag)	0.255	0.260	0.250	0.185
S3 maximum (tree-local bag)	0.080	0.080	0.070	0.060
ϕ_∞, *complete* fingerprint ($\mathcal{X}$)	0.025	0.025	0.020	0.005

AI. UNSEENEQCLASS AT OTHER k, AND THE TIE-BREAKING RULE (TAGS `r71_tier3_baselines_v3`, `r78`)

Two questions a reader should ask of Table 26: is the ordering an artefact of k=5, and is k-NN class precision even well defined for a representation that maps many forms to one vector? Answering the first exposed a defect behind the second, so we report both, together with the corrections they force on our own previously reported numbers.

k-robustness, and a second thing we had wrong. Table 30 sweeps $k \in \{1, 3, 5, 10\}$ at every scale. The trained encoder exceeds the strongest non-learned baseline *of this table* in *every* one of the 20 cells, so no conclusion in §4.2 depends on the choice; at K=500 it also leads the *searched* composite of Appendix BB at all four k, which is a stronger statement at that one scale and is not available at the others. The absolute values fall steeply in k for every representation, and that is a property of the *metric*: a query's class supplies at most $\min(m_c-1, k)$ of its k neighbours, so the attainable maximum itself falls in k, 1.000/0.969/0.908/0.723 at k=1/3/5/10 against the

encoder's $1.000/0.965/0.894/0.710$, which is 96–100% of attainable at every k and every K (Appendix Z; the unseen pool at $K{=}500$ holds a median of 6 forms per class, mean 7.3, range 2–20). **The decay is that maximum descending with the encoder tracking it, not the encoder degrading.** The explanation earlier drafts gave for it was wrong and its correction *strengthened* the result (`REVISION_HISTORY.md` §7); what defends $k{=}5$ is that it was fixed before these sweeps ran and that every k gives the same verdict against its own attainable maximum.

Table 30: **k-robustness of UNSEENEQCLASS** on `poly8`. The trained encoder leads the strongest non-learned baseline in all 20 cells. Values fall with k for every representation because the metric's own *maximum* falls with k (a class of m forms supplies at most $\min(m{-}1, k)$ class-mates): $1.000/0.969/0.908/0.723$ at $k{=}1/3/5/10$, which the Tree-LSTM row tracks at 96–100% throughout. Read each column against its attainable maximum, not against 1.000.

		$k{=}1$	$k{=}3$	$k{=}5$	$k{=}10$
Tree-LSTM	$K{=}50$	1.000	0.939	0.843	0.625
	$K{=}100$	0.991	0.922	0.882	0.728
	$K{=}200$	0.995	0.937	0.872	0.707
	$K{=}500$	1.000	0.965	0.894	0.710
	$K{=}1000$	0.995	0.948	0.868	0.642
Bag (full token)	$K{=}50$	0.984	0.887	0.793	0.493
	$K{=}100$	0.836	0.807	0.770	0.645
	$K{=}200$	0.804	0.730	0.673	0.554
	$K{=}500$	0.580	0.547	0.517	0.444
	$K{=}1000$	0.586	0.549	0.509	0.418
Bag (tree-local, S3)	$K{=}50$	0.839	0.790	0.726	0.447
	$K{=}100$	0.743	0.708	0.649	0.565
	$K{=}200$	0.559	0.518	0.492	0.418
	$K{=}500$	0.295	0.283	0.277	0.256
	$K{=}1000$	0.286	0.298	0.277	0.256

The metric is ill-defined for a bag, and our own implementation was wrong. A bag collapses many distinct forms onto one vector, exactly the property the S2 argument rests on, so a bag's neighbour list contains long runs of *exactly equal* similarities and which of them fall inside the top k is arbitrary. At $K{=}500$, $k{=}5$ the k-th neighbour is on average 14.7-way tied for the full-token bag and 93.0-way tied for the variable-blind bag. Three consequences, in increasing order of severity. First, `torch.topk` leaves the choice unspecified, so the top 5 of a `topk(10)` call need not equal `topk(5)` and the sweep above was not internally consistent. Second, in `float32` the ties are not even *detected*: mathematically identical similarities differ in their last bits according to the order the feature vocabulary happens to be laid out in, so a stable sort ranks by floating-point noise, and that noise is correlated with vector content rather than being blind to the labels. Third, the effect is large, not cosmetic: the WL $h{=}1$ bag scored 0.276 or 0.424 purely according to whether its vocabulary was built over the unseen forms or over the whole corpus, a change that cannot alter any similarity.

The rule we adopt: accumulate similarities in `float64`, round to 10^{-6}, then rank with a stable sort under one fixed random permutation of the candidates. This is deterministic, prefix-consistent across k (so the sweep is coherent), and blind to the labels. It is also unbiased across classes, which breaking ties by raw index would not be, since candidates are laid out in class order. Rounding is safe because the gap spectrum is cleanly bimodal: across all baggers and all scales, floating-point residual gaps top out at 1.9×10^{-7} while the smallest genuine gap is 1.3×10^{-5}, a valley spanning $69\times$ with the threshold inside it. Rather than trust a measurement made once, the code re-derives this on every similarity matrix it ranks and warns if any gap lands within a factor of four of the threshold while the k-th neighbour is tied, the combination under which rounding could change a ranking it has no business changing. Under the corrected rule two independently written implementations (`r70`'s and `r71`'s, which build their vocabularies differently) agree to all four reported digits at all five scales; before it they did not.

Corrections this forces, and their direction. Table 26 and Table 27 now report every non-learned row from a single process under this rule, which moves several values we previously published. The strongest non-learned baseline, the number our comparisons actually rest on, barely moves

($0.518 \rightarrow 0.517$ at $K{=}500$). The heavily tied S3 rows move a lot and mostly *downward* (tree-local $0.411 \rightarrow 0.277$, best WL/ϕ_d member $0.383 \rightarrow 0.276$ at $K{=}500$), and the variable-blind bag moves in both directions across scales ($0.535 \rightarrow 0.561$ at $K{=}50$, $0.186 \rightarrow 0.146$ at $K{=}500$). We flag the direction explicitly because most of these corrections favour our own claim, which is a reason for more scrutiny rather than less: the justification for the new rule is that it is invariant to a change that cannot affect any similarity, and that justification was fixed before the corrected values were computed. One substantive claim weakens as a result. We previously wrote that S3 statistics account for more than half of the distance from the operator bag to the strongest bag; under the corrected numbers the share is roughly a third. The trained rows are unaffected, for a measured reason rather than an assumed one: the encoder's k-th neighbour is 1.0-way tied, all 50 random tie orders return the identical score, and the optimistic and pessimistic tie bounds coincide (tag `r78`, `--with_encoder`).

How much is tie luck? The bags are degenerate, and the oracle bound is not a baseline. Under the reported rule the baselines are stable: over 50 random tie orders at $K{=}500$, $k{=}5$ the full-token bag spans 0.503–0.526 and the variable-blind bag 0.129–0.152, reproducible to about ± 0.01, and the swing tracks tie multiplicity, ± 0.008 for the token n-gram (10.9-way) through ± 0.015 for the S3 rows (24-way) to ± 0.022 for the variable-only bag (61-way), the loosest row we report. The *achievable* range is another matter: resolving each tie to maximise or minimise precision gives $[0.354, 0.742]$ for the full-token bag and $[0.017, 0.798]$ for the variable-blind bag. A skeptic could quote that 0.798 as nearly matching the encoder, so we say plainly why it is not a baseline, attaining it requires ordering tied neighbours *using the class labels*, the quantity being predicted. The width measures degeneracy: with 93 candidates tied at the fifth neighbour the variable-blind bag does not rank at all here, and its score is set almost entirely by the tie convention. That is a stronger form of the S2 argument than the point value, and the reason a bag's k-NN number should never be quoted without one.

The same defect in the composition protocol, and what it does to that claim. The correction above is confined to the k-NN routine, which only three scripts call; the composition experiment of §4.3 instead identifies a held-out composite by `argmax` over class centroids. That is a different computation, so we measured it rather than assuming it was safe (tag `r78`, `--r76_bags`), and it is *not* safe: at the top of each composite's similarity list the true class is tied with between 1.1 and 45.2 other centroids depending on the bag, and `argmax` silently returns the lowest index. Two runs of that deterministic computation in one environment return *different* scores for the WL bags (0.100 vs. 0.095 at $h{=}2$, 0.105 vs. 0.110 at $h{=}3$) while the corpus itself hashes identically across processes, the `float32` defect again, in a protocol that never touches the k-NN code. Re-scoring the whole zero-parameter ladder under the label-blind rule (`float64`, quantised, one fixed permutation) is reproducible across runs and moves the WL/ϕ_d family *down* again, to 0.060–0.085 from 0.080–0.110 on the INVASIVE pair. That correction buys the paper nothing, and we say so rather than let the smaller range be read as the new headline: the S3 *maximum* is set by the tree-local bag, which is only 1.25-way tied and does not move at all, so the INVASIVE column's S3 maximum stands at 0.110 in the unnumbered ladder above, which is where that number lives, not in any numbered table. The body no longer quotes it at all: §4.3's non-learned figure is `r93`'s own 0.100 at $d{=}8$ (Table 33), a later run whose ladder this correction does not touch, and there the strongest non-learned member is the *variable-only* bag, not the tree-local one, which reads 0.035 in that table. Both the $d{=}8$ value and *which member attains it* are asserted against the stored log, so the substitution is checkable rather than asserted here. Appendix AH gives the re-scored ladder for both pairs.

The composition claim is unaffected in substance. The strongest non-learned baseline is the token n-gram at 0.135, only 1.1-way tied with oracle bounds $[0.115, 0.140]$; no bag with fewer than three tied centroids has an optimistic bound above 0.205; and the untrained-encoder control is tie-free here too (tie bounds of zero width). Against the primitives-only arm's 0.993 none of that is close. The two variable bags are the exception, named rather than omitted: at 17.5- and 45.2-way ties their optimistic bounds reach 1.000 and 0.530, degeneracy again, not capability, and on a S1-immune corpus a variable bag carries almost no information by construction. The twin probes need no such audit: there every bag is pinned at *exactly* 0.500 by construction, the same degeneracy reported rather than hidden.

The same defect once more, in the one place we had claimed to have covered. §4 asserts that *every* similarity-based evaluation in this paper breaks ties label-blind. That was false for the depth table, whose ladder still used raw `float32 argmax`, and it went unnoticed *because* the claim was already written; a cross-architecture assertion added for a different purpose caught it (`REVISION_HISTORY.md` §6). Re-scoring all four depths under the label-blind rule (tag `r78_r79_bag_ties`, `--r79_bags`) moves exactly two of the cells Table 28 prints, each by 0.005: the S3 maximum at d=4 from 0.050 to 0.055, and ϕ_∞ at d=3 from 0.030 to 0.025. Nothing else moves, the strongest non-learned baseline is the full-token bag at every depth under both rules and identical to three decimals, so the $2.9\times$ and $8.2\times$ ratios and the $38\times$ over ϕ_∞ stand, as do the two family-level assertions that carry the argument: $\phi_{d\geq 4}$ still equals the complete order-blind fingerprint exactly, and the complete *ordered* fingerprint still scores below the coarse tree-local bag at every depth. We do *not* add this to the four-claim tally of §1: it is not a fifth defect but the third one incompletely fixed, found in a protocol we had already audited and then re-imported. **A correction is not done when the numbers are re-run; it is done when every path that can reach the defect is enumerated**, and a universal claim in the prose is the easiest place for that enumeration to go stale.

Enumerating the path in advance, for once (tag `r78_r81_bag_ties`, `--r81_bags`). The primitive-set run of §4.3 imports `_bag_identify` from `r76` for the same reason it imports everything else, so its zero-parameter ladder is the *published* protocol rather than a re-implementation, and therefore inherits the same `float32 argmax`. This time that was known before the run rather than discovered after it, so the re-scoring mode was written alongside the experiment and Table 29 quotes the label-blind values from the start. The correction is larger here than on the depth curve: seven cells move rather than two, by up to 0.020 (the variable-blind bag at d=3, $0.030 \rightarrow 0.050$), and in *both* directions, the full-token bag gains 0.010 at d=1 and loses 0.005 at d=3. No ordering changes: the full-token bag is the strongest non-learned baseline at every depth under both rules, the S3 maximum is the tree-local bag at every depth, and the $3.9\times$ and $5.0\times$ ratios the section quotes are computed from the re-scored row. We note the direction mix rather than only the magnitude, because a correction that moved every cell the same way would be a bias and this one is not.

AJ. Is the Flat Depth Curve the Protocol's, or the Tree-LSTM's? (tags `r79_depth_gnn`, `r79_depth_transformer`, and both `_rep2`)

Table 28 shows a Tree-LSTM holding 0.945–0.967 across composition depths 1–4. Read carelessly, that says depth-robust composition is what the *protocol* elicits. It is not, and the cheapest way to see so is to swap the encoder and change nothing else: because the never-composed arm trains once per seed and is evaluated at every depth, a second architecture costs five trainings. Everything below is the same corpus, the same 200 classes, the same library, the same held-out chains, the same 5 seeds and the same 10 epochs; only the encoder differs, and `verify_claims.py` asserts that field by field against the Tree-LSTM run's own provenance rather than against a copy of it.

Two replicates per architecture, because one would have been misleading. We report each architecture twice, as a range, for a reason that is itself a finding: **GIN and Transformer training in this harness is not run-to-run reproducible, and Tree-LSTM training is bit-for-bit *within a session*.** Two back-to-back invocations of the depth sweep (same seeds, same data, same flags, same process environment) give the Tree-LSTM identical numbers to three decimals at every depth, while GIN's five-seed means move by up to 0.013 and the Transformer's by up to 0.101. The means understate it: the *per-seed* numbers behind them differ by up to 0.065 and 0.375, so a mean is stable here only to the extent that five seeds average over the same nondeterminism. The untrained controls and the whole non-learned ladder are exactly reproducible for all three, so this is not the corpus and not the protocol: it is nondeterminism in the scatter-reduction and attention kernels these two encoders use on this accelerator, which no seed controls. Had we run each architecture once and quoted three decimals, the table would have looked like the others in this paper and been a different kind of object.

Encoder, never composed	d=1	d=2	d=3	d=4	d=1 minus d=4, paired
Tree-LSTM *(reproducible)*	**0.965**	**0.967**	**0.961**	**0.945**	+0.020 [0.008, 0.032]
GIN	.935–.937	.781–.788	.781–.787	.637–.650	+0.285 to +0.300
Transformer	.741–.790	.600–.663	.520–.621	.423–.426	+0.318 to +0.364
Controls, each exactly reproducible within its architecture					
Untrained Tree-LSTM	0.427	0.248	0.278	0.153	—
Untrained GIN	0.637	0.095	0.095	0.027	—
Untrained Transformer	0.316	0.116	0.116	0.036	—
Strongest non-learned (all three)	0.330	0.190	0.190	0.115	—

The bags are encoder-independent by construction and identical, to the last digit, across all five runs, with one exception we name rather than round away: the three `hash()`-based WL rows disagree between runs by 0.005–0.010, the `PYTHONHASHSEED` process-dependence documented in Appendix AI. It was that disagreement, surfacing in an assertion written to check something else, that exposed the depth table's un-re-scored tie rule, so we leave it visible. Two conclusions follow, and they pull in opposite directions.

Against generality. The near-flat curve is the Tree-LSTM's alone. Over the same four depths GIN loses 0.285–0.300 and the Transformer 0.318–0.364, every replicate's paired interval far from zero, and every one at least 14× the Tree-LSTM's 0.020, a separation so wide that the irreproducibility above cannot touch it, which is the only reason a range is enough here. So §4.3's positive result is scoped to an architecture with explicit recursive composition, and any reading of it as "held-out-form protocols elicit depth-robust composition" is unsupported. We say so in the body rather than leaving the qualifier to an appendix, once, and we state where: §4.5's scope paragraph, *never architecture-independent*, which routes here. §5 scopes the same claim by class count and by algebra but *not* by architecture, so this is the one place the body carries it.

For the criterion. Every architecture still clears the whole non-learned ladder *and* its own untrained control at every depth, in both replicates. The worst cell in the table, the Transformer's 0.423 at d=4, is 3.7× the strongest non-learned baseline and 12× an untrained Transformer. The S3 requirement of §3.2 is therefore satisfied by all three architectures at all four depths *on this basis*, Appendix AK widens it to four primitives and the Transformer stops satisfying it past d=2, so what follows is a property of the narrow design and not of the criterion. On the narrow design it is exactly the point of insisting that the criterion be *necessary and not sufficient*. Clearing it does not rank these three encoders, and the property that distinguishes them, how the margin behaves as composition deepens, is invisible to a single-depth comparison however many baselines that comparison beats. This is the sharpest instance in the paper of its own thesis, and it argues that depth, not baseline count, is the axis worth adding to a held-out-form protocol.

Why the untrained control has to be swapped too, and what it cost us to forget. The untrained rows above differ enormously between architectures, at d=1 an untrained GIN reaches 0.637 where an untrained Tree-LSTM reaches 0.427, and by d=2 the GIN has collapsed to 0.095, *below* every bag. So the floor a trained row must clear is architecture-specific, and quoting one architecture's untrained control against another's trained row would misstate the margin in either direction by up to 0.3. An earlier version of this table quoted a control that had not been swapped with `--arch`; the verifier caught it as an equality that held when it should not have, and the fix moved the trained numbers too (`REVISION_HISTORY.md` §8). **The pattern is general: when a comparison is built by swapping one component, the controls are part of what has to be swapped**, and a control that silently does not move is indistinguishable from one that correctly does not move unless something asserts which.

This table's design has no composites-in-library arm, so it carries no upper bound and no paired cost-of-never-composing; the log records that omission rather than leaving it to be inferred. Whether GIN and the Transformer *could* do depth-4 composition when trained on composites is a different question from the one asked here, and it is not left open: Appendix AK runs that arm for both encoders on the harder four-primitive design, so what is missing here is this narrow design's upper bound, not the answer.

AK. The Same Question of the Four-Primitive Curve, and a Different Answer (tags r81_depth_gnn, r81_depth_transformer, and both _rep2)

Appendix AJ asked whether the flat depth curve was the protocol's or the Tree-LSTM's, on r79's single path over two schemas. Table 29's curve is a harder object (four primitives, all 24 orders) so the question has to be asked of *it* rather than inherited from a narrower design, and the answer is not the same answer twice. Everything below is the same corpus, the same 200 classes, the same libraries, the same held-out chains, the same 5 seeds and the same 10 epochs; only the encoder differs. verify_claims.py asserts that field by field against the Tree-LSTM run's own provenance rather than against a copy of it, and asserts the *construction* identical too (200 classes, all 24 orders, mean test-form token counts 16.88/21.08/24.94/28.70) because the sampling that builds the depth chains runs before the encoder exists and nothing upstream of the encoder may move.

Two replicates per architecture, for Appendix AJ's reason. GIN and Transformer training in this harness is not run-to-run reproducible and Tree-LSTM training is bit-for-bit within a session (Appendix AL bounds what that does *not* cover). Two back-to-back invocations move GIN's five-seed means by at most 0.010 and the Transformer's by 0.039; the *per-seed* values behind those means move by 0.050 and 0.155. Both untrained controls are exactly reproducible, and here every bag agrees to the last digit across all five runs, the hash()-salted WL rows included, which they need not have, so the verifier asserts the ± 0.010 bound rather than tightening to the equality this particular set of processes happened to give.

Encoder, never composed	d=1	d=2	d=3	d=4	d=1 minus d=4
Tree-LSTM *(reproducible)*	**0.997**	**0.983**	**0.950**	**0.918**	+0.079 [0.063, 0.097]
GIN	.927	.863–.873	.777–.781	.667–.669	+0.258 to +0.260
Transformer	.648–.654	.397–.422	.231–.270*	.135–.167*	+0.481 to +0.519
Controls, each exactly reproducible within its architecture					
Untrained Tree-LSTM	0.482	0.379	0.289	0.215	—
Untrained GIN	0.361	0.207	0.117	0.074	—
Untrained Transformer	0.220	0.211	0.150	0.068	—
Strongest non-learned	0.255	0.260	0.250	0.185	—

*the two cells where the trained encoder does *not* clear the strongest non-learned baseline on its class-level interval, in *either* replicate. The non-learned row is the label-blind one Table 29 prints; these logs carry r76's float32 argmax, whose row differs by up to 0.010 at $d \leq 3$, and the verifier asserts that every verdict below is the same under both rules, a S3 verdict that turned on a tie rule would be an artefact of the tie rule.

Against generality, and more sharply than on the narrow basis. The price of composition is the encoder's, and the three architectures order strictly: +0.079 [0.063, 0.097] for the Tree-LSTM, +0.258 and +0.260 for GIN, +0.481 and +0.519 for the Transformer, 3.3× and 6.1–6.6× the Tree-LSTM's decay, with *no* replicate's interval touching another architecture's, so the ordering is resolved over classes rather than read off point values. Each curve is monotone non-increasing. So §4.3's composition result is scoped to an encoder with explicit recursive composition, and the wider basis makes that scoping tighter rather than looser.

And this time the criterion bites. Appendix AJ could report that all three architectures cleared the whole non-learned ladder at all four depths, necessary-and-not-sufficient illustrated by a criterion nobody failed. Under the four-primitive design that is no longer true. **Trained on primitives alone, the Transformer passes the $\mathcal{F}_3$ audit at depths 1 and 2 only.** At d=3 the strongest non-learned baseline lies *inside* its class-level interval in both replicates (0.270 [0.239, 0.301] and 0.231 [0.204, 0.258] against 0.250), so the comparison is undecided; at d=4 both point values fall *below* the baseline (0.167 and 0.135 against 0.185), and in the second replicate the baseline sits above the whole interval [0.112, 0.159], a zero-parameter bag of tokens resolvedly beating a trained Transformer. By Definition 1 the Transformer's depth-3 and depth-4 composition results do not pass the $\mathcal{F}_3$ audit, and by Proposition 2 no axis C claim follows from them. Nothing about the Tree-LSTM or GIN changes: both pass the $\mathcal{F}_3$ audit at all four depths in both replicates, each clearing its *own* untrained control on the interval. Two further invocations of this row arrive below with the compos-

ites arm, and they qualify this verdict rather than repeat it: d=4 is refused by all four, while at d=3 one of the four clears, its interval reading $[0.262, 0.324]$.

Why the interval clause in Definition 1 is load-bearing rather than decorative. At d=3 a point-value audit would have licensed the Transformer on the first replicate ($0.270 > 0.250$) and refused it on the second ($0.231 < 0.250$): the same experiment, run twice, opposite verdicts. Requiring the margin to be resolved over classes is what makes the verdict replicate-invariant here, and it is not a stylistic preference, on an architecture whose per-seed values move by 0.155 between identical invocations, a criterion stated on point values is a criterion that reports the accelerator. This is the one place in the paper where our own criterion declines to license a result we ran, which is the behaviour we would want from it and the reason §3.2 states it as an object rather than as advice.

Order is still not the fragile axis, for either encoder. Ten of the twelve architecture-replicate-depth cells put the length-identical alternate order at or above the forward chain; the two exceptions are Transformer cells at -0.005 and -0.002 whose intervals contain zero, so no replicate resolves a deficit. The d=4 per-order spread is $[0.520, 0.840]$ for GIN and $[0.000, 0.280]$ for the Transformer, whose weakest of the 24 orders reaches exactly 0.000 in the second replicate over the $\sim$3 classes drawing it, printed because a mean over orders is precisely where that would hide, and noisy for the reason the log states rather than for a reason we chose.

The untrained control, swapped again. At d=1 an untrained Tree-LSTM reaches 0.482, an untrained GIN 0.361 and an untrained Transformer 0.220; by d=4 they are 0.215, 0.074 and 0.068. The floor a trained row must clear is architecture-specific by up to 0.26, so §3.2's S3 requirement is not one number, and the verifier asserts the three rows *differ* rather than asserting values alone, the shape that caught the defect Appendix AJ records, where `--arch` reached training and not the control. The composites-in-library arm was missing for these two encoders when this appendix was first written, and its absence was recorded here rather than left to be inferred; it is the table below, so these rows now carry both an upper bound and a paired cost of never composing.

The arm this appendix said it had not run, and the verdict it moves. The rows above are Table 29's *first* row for two further encoders; below is the rest of that design for them, pre-registered before it was run, including which outcome licenses which sentence, because it could go either way. Four invocations, two per encoder: the Tree-LSTM run's own recorded arguments with `--arch` changed and `--primitives_only` dropped. No rewrite schema was written, so `equivalence.py` is untouched and no number already in this paper can move. `verify_claims.py` asserts `construction` and `closure_guard` equal to the Tree-LSTM run's *as whole dictionaries*, the untrained control equal to this appendix's *per seed*, and every non-learned bag equal to the last digit, so the floor each new row is read against is the published one, and dropping a flag has not moved the control's place in the RNG stream.

Library, tested at	d=1	d=2	d=3	d=4	closes	pays
GIN (tags `r81_matrix_gnn`, `_rep2`)						
primitives only	.925/.927	.844/.870	.755/.771	.632/.679	—	—
∪ depth-2	.932/.918	.871/.859	.783/.762	.699/.686	—	—
∪ depth-3	.918/.919	.855/.866	.792/.793	.708/.706	—	—
∪ depth-4	.891/.892	.832/.833	.770/.778	.697/.711	—	—
paired gain				$+.065/+.032^{\ddagger}$	22%/13%	3/4
Transformer (tags `r81_matrix_transformer`, `_rep2`)						
primitives only	.645/.643	.404/.437	.259/.293	.162/.180	—	—
∪ depth-2	.635/.645	.678/.666	.479/.489	.375/.390	—	—
∪ depth-3	.500/.518	.495/.502	.619/.606	.456/.466	—	—
∪ depth-4	.498/.527	.342/.359	.331/.327	.482/.472	—	—
paired gain				$+.320/+.292$	66%/63%	4/3

closes is the fraction of that encoder's *own* d=1 minus d=4 deficit that its full library recovers (the paired gain over the paired row-1 drop, on the same classes) so each encoder is read against its own ceiling rather than the Tree-LSTM's. *paired gain* is the full library's row minus row 1 at the same test depth, the paper's cost of never composing, so comparable to the Tree-LSTM's $+0.069$; where a *shallower* library scores higher at d=4 (GIN rep 1, $+0.076$ at training depth 3) the matrix

above shows it and this row does not fold it in. *pays* counts how many of the 6 above-diagonal cells fall more than 0.005 *below* row 1 at their own test depth: accuracy given up at shallower depths to buy deeper ones. The Tree-LSTM's pair is 87% and **0** on `poly8` (Table 29) and 13% and 5 on `boolean8` (Appendix AM). ‡this interval contains zero ($[-0.004, 0.066]$), so in that replicate GIN's gain is not resolved at all; the Tree-LSTM's is $+0.069$ $[0.052, 0.087]$.

What the arm shows. **Composites in the library recover part of the depth-4 deficit for every encoder; only for the Tree-LSTM on `poly8` is the recovery free.** That is the *pays* column, and it is the one signature stable across both replicates of both new encoders: **0** of 6 for the Tree-LSTM against 3–4 for GIN *and* 3–4 for the Transformer, and the algebra swap that already broke the closure breaks it the same way (`boolean8`, 5 of 6). So the object §4.3 reads as a coverage deficit, a gap the library closes at no cost elsewhere, is that encoder's on that corpus, which is what the bold clause there now says. On *closes* alone the encoders do not even order as the decay above does, which is the second reason the clause names a mechanism rather than a ranking. The loud reading; that GIN's test-depth-4 column is non-monotone in training depth, as `boolean8`'s is, was drafted and withdrawn: it is not replicate-stable (argmax at training depth 3, then 4), so no sentence here rests on it.

And the audit verdict above is a verdict about an exposure, not about an architecture. The bolded restriction earlier opens *trained on primitives alone*, and that clause is this arm's doing: it was absent until these runs, and without it the sentence asserted of the encoder what had been measured of one exposure. Measured on the interval, never on a point value: the never-composed row is refused at d=4 by all four Transformer invocations and at d=3 by three of them, the fourth clearing by 0.007, thinner than the replicate spread, so we report three of four rather than a universal, and it is this appendix's own *same experiment, opposite verdicts* recurring at the criterion's boundary. The library matched to the test depth clears those same bags at all four depths in *both* replicates, lower bounds .613/.650/.586/.449 and .613/.636/.573/.439 against .245/.255/.255/.185. This is the sharpest form of §3.2's claim that the audit licenses a claim about a *trained model on a task*: what the model was trained on can move the verdict with the encoder held fixed, so an architecture-level reading of either verdict is unlicensed in both directions.

The free replication, and the envelope it widens. Training depth 1 *is* the never-composed arm, so these runs re-run the rows above at no extra cost, registered in advance as a bound that could widen. It did: GIN's paired d=1 minus d=4 reads $+0.258$, $+0.260$, $+0.293$ and $+0.248$ over the four invocations, and its per-depth row-1 envelope is 0.047, not the 0.010 two invocations gave. 0.047 **is the envelope; the figure above is the two-invocation special case**, which is why both stand rather than one replacing the other; that sentence is about back-to-back invocations and is true of them. The ordering survives the widening: 0.047 is under a third of the 0.169 gap to the Tree-LSTM's curve. It widened for *both* encoders, which is why the check was run on both: the Transformer's four-invocation row-1 envelope is 0.062 against the 0.039 two gave, its paired drops reading $+0.481$, $+0.519$, $+0.483$ and $+0.463$. And the composites arm's *own* replicate envelope, measured here rather than inherited from the never-composed arm, since a bound on one arm is not a bound on another, is 0.047 again (GIN, coincidentally) and 0.034 (Transformer) over the 16 matrix cells.

AL. Is the K=500 Result a Property of One Class Partition? (tags `r82_split*`, `r82_ladder*`, `r82_null*`)

Every `poly8` number in this paper is read off *one* class partition. UnseenEqClass splits classes by index from a single `Random(70)` shuffle taken before training (Appendix Z), and the 5 training seeds we report vary the initialisation *within* that partition. Training randomness and partition randomness are different sources of uncertainty, and only the first was ever measured: no number of training seeds says anything about the choice of split. So the split seed became a flag, `--shuffle_seed`, threaded identically through `run_r70`, `run_r71` and `run_r80`, and the K=500 cell was re-run over five independent partitions, seed 70 being the published one.

The flag is inert at its default, and moves the split away from it. Under `--shuffle_seed 70` the non-learned ladder reproduces the published one across all 583 scalar cells of the K=500 block, exactly. Under seed 71 the partition genuinely moves: the unseen pool holds 793 forms rather than 731, its exact null is 0.01419 rather than 0.01372, and the strongest non-learned baseline reads 0.5498 rather than 0.5174. Both directions matter. The first says the flag did not silently perturb

what the paper already reports; the second says the seed reaches the shuffle rather than merely the command line, and note that *the number of unseen classes cannot detect this*, since it is pinned at $\min(K/5, 100)$=100 for every split. The verifier therefore asserts the inequality on each partition's exact null, not on its class count.

An honest boundary on Appendix AJ's reproducibility claim, found here. Re-running the K=500 Tree-LSTM cell under the current code returns unseen 0.8966 where the shipped log records 0.8943, with *every* per-seed value differing (0.8955/0.9007/0.8906/0.8963/0.8999 against 0.8947/0.8963/0.8862/0.8977/0.8969). The partition is not what moved: the non-learned rows shift by at most 0.0005, where a genuinely different partition moves them by 0.03. Nor is it the tie rule, which leaves trained rows untouched by construction (Appendix AI). The two runs are separated by months and by an interpreter upgrade, and we did not spend a further 3.5 GPU-hours to separate cross-environment drift from run-to-run nondeterminism. What it establishes is a *scope*: Appendix AJ's "Tree-LSTM training is bit-for-bit" is a statement about two back-to-back invocations in one process environment, which is how it was measured, and **not** a claim that a trained cell reproduces exactly across environments. The envelope observed across environments is 0.0023, an order of magnitude below the effect any claim in this paper rests on, and stated rather than left for a reader to discover.

Five partitions, and what moves between them. Each row is an independent `Random(s)` shuffle of the 1102 classes, then the same index split (TRAIN = first 500, UNSEEN = next 100) and the same 5 training seeds. The unseen *class* count is pinned at 100 by the $\min(K/5, 100)$ cap for every split, but the unseen *form* count is not, and neither is the class-size multiset that sets each partition's exact null and its attainable maximum, so both are recomputed per split and each row is read against its own.

Split	forms	null	max	Tree-LSTM (95% CI)	strongest bag (95% CI)	of max
70 *(published)*	731	.0137	.908	.897 [.860, .925]	.517 [.435, .593]	.988
71	793	.0142	.908	.903 [.866, .931]	.550 [.461, .629]	.994
72	683	.0145	.878	.875 [.831, .910]	.524 [.431, .608]	.997
73	666	.0145	.879	.872 [.825, .907]	.439 [.358, .517]	.992
74	737	.0137	.901	.896 [.856, .926]	.494 [.408, .573]	.994
mean ± sd	—	—	.895±.015	**.888±.014**	**.505±.042**	**.993±.003**

The result survives, and the published split is not a lucky one. The trained encoder's class-level interval clears the strongest non-learned baseline's in **all five** partitions, and the published values sit inside the observed ranges rather than at their edges: 0.894 against $[0.872, 0.903]$, and 0.517 against $[0.439, 0.550]$. The *full-token* ratio is if anything conservative, over five partitions it is 1.77 ± 0.14, its smallest value 1.64. **It is no longer the ratio the body prints**: §4.2 now reads its margin against the *searched* composite of Appendix BB, 1.3×, and **this sweep's own rows do not cover that line**, what they re-ran is the *catalogued* baseline, so the split sensitivity these five partitions bound is its, not the search's. **The searched line is covered separately, and by these same five partitions**: the composite is frozen after selecting on 70 and re-scored on all four of the others, where the trained rows below supply the encoder it is read against (Table 36). It stays below the encoder in every partition, and it is *more* split-sensitive than the catalogued baseline it replaced rather than less, that table prints both dispersions over one common set of partitions, which is the reason the check is worth running on four splits rather than the one it began with. Corollary 2's non-monotonicity also holds in every partition: ϕ_∞ (0.224 ± 0.019) scores below the coarsest *declared* member $\phi_{d=1}$ (0.266 ± 0.023) five times out of five, so that claim was not an artefact of one draw either. (The tree-local bag reads 0.266 ± 0.023 as well, to three decimals; we quote $\phi_{d=1}$ because Definition 1 enumerates it and the corollary is about the family.)

What the partition actually perturbs is mostly the attainable maximum, and the baseline. Two numbers make this concrete. First, **the non-learned baseline is three times more split-sensitive than the encoder**: the strongest bag moves ±0.042 across partitions against the Tree-LSTM's ±0.014, its range spanning 0.439–0.550. A held-out-form margin is therefore a statement about which classes were drawn to a degree the baseline, not the model, controls. Second, **most of the encoder's own variation tracks that maximum**: expressed as a fraction of each partition's own attainable maximum the Tree-LSTM reads 0.993 ± 0.003, an SD 4.3× smaller than the raw ±0.014. Split 72 scores

lowest in absolute terms (0.875) because its attainable maximum is lowest (0.878), not because the encoder did worse on it; it is that split's *best* row at 0.997 of attainable. Had we reused seed 70's ceiling, as a sweep that recomputed the split but not its null would, that partition would have read as a 0.022 drop in encoder quality instead of a 0.030 drop in what *any* representation could reach.

Cost and scope. Five trained partitions at $K{=}500$ cost 895 GPU-minutes here. This bounds the split sensitivity of the $K{=}500$ cell only; the $K{=}50$ and $K{=}100$ rows of Table 26, where the intervals already overlap on the published split, are not re-run and we do not claim their overlap is partition-independent.

AM. Does the Composition Design Replicate Off `poly8`? Two Further Corpora, One of Them a Different Algebra (tags `r83_composition_onevarpoly13`, `r84_composition_boolean8`)

Table 29's anchors, equivalence classes and variable pool are EQNET's (Allamanis et al., 2017), but the *rewrite chains* are ours, so the composition result, unlike everything else in the paper, rested on one corpus and one algebra. That is the same narrowness we held against `r79`'s two-schema path (§4.3), one level up. The design was therefore re-run on two further EQNET corpora with nothing changed but `--corpus`: the recipe, the epochs, the seeds, the guard, the metric and the non-learned ladder are the code paths that produced Table 29, and `poly8` remains the flag's default so no shipped number could move.

What each corpus contributes, and why the second one is the harder test. `oneVarPoly13` (677 classes) is the *same* algebra on different anchors: the four `poly8` primitives port unchanged. `boolean8` (232 classes) is a *different* algebra, and the `poly8` primitives are invalid on it; they inject arithmetic operators into boolean trees, which is why a naive feasibility probe reports three of the four as firing there. It needed four new ones, held to the same two constraints the original four were measured against: each must be an exact identity, and each must have a *site-independent* token delta, since the paired-depth columns and the size-matched alternate-order arm both require every depth-d form to be one fixed size over its anchor for all $4!$ orders.

Corpus	Algebra	Primitives (token delta)	Anchors supporting all four
`poly8` *(published)*	polynomial	`commute` (0), $\neg\neg$ (8), $x{+}0$ (4), $x{\cdot}1$ (4)	186/200
`oneVarPoly13`	polynomial	the same four, unchanged	199/200
`boolean8`	boolean	$\wedge/\vee/\oplus$ commute (0), $\neg\neg$ (6), De Morgan (9), $x{\wedge}\top$ (4)	193/200

The vocabulary change `boolean8` required was made so that *no published number could move*: `VOCAB_SIZE` stays frozen at its pre-boolean value of 34 and the five boolean tokens are *appended* (to 39), so every existing token id is unchanged and every arithmetic encoder in this paper is built over the same vocabulary as before.

Construction, verified per run rather than inherited. `oneVarPoly13`: $K{=}200$ kept from 205 scanned (3 anchors with no site for some primitive, 2 where the leading primitive needed two sites); all 24 orders represented; every depth-4 form exactly $+16$ tokens over its anchor *for all* 24 *orders*; 800/800 token-delta pairs equal to the sum over the drawn primitives, no exceptions; mean test-form tokens 28.56/32.76/36.70/40.62. `boolean8`: $K{=}180$ kept from 210 scanned (6 no-site, 19 leader-needs-two-sites, 1 composite and 4 order-arm failures); all 24 orders; every depth-4 form exactly $+19$ tokens for all 24 orders; 720/720 delta pairs; mean tokens 19.82/24.74/29.56/34.52. The independent-seed closure guard, re-searching each anchor's $(d{-}1)$-application closure under seeds disjoint from construction's, finds 0 test forms, 0 training composites and 0 alternate-order forms inside the closure, on 200/200 and 180/180 classes, at every depth. Together the two runs cost 5.2 GPU-hours here.

oneVarPoly13, K=200 of 677 classes, 5 seeds, chance 0.005				
training depth ↓ / *test* depth →	d=1	d=2	d=3	d=4
1: **composition never seen**	**.995**	**.979**	**.931**	**.849**
95% CI over classes	[.99, 1.0]	[.97, .99]	[.91, .95]	[.83, .87]
2: depth-2 composites in library	.998	.991	.975	.936
3: depth-3 composites in library	.999	.991	.986	.973
4: depth-4 composites in library	.999	.997	.988	.979
cost of never composing, paired	—	+.012	+.055	**+.130**
Alternate order, same multiset	—	.980	.926	.845
Untrained encoder ($\mathcal{X}$)	.377	.320	.221	.174
Strongest non-learned	.810	.695	.435	.365
S3 maximum	.810	.695	.435	.365
ϕ_∞, *complete* fingerprint ($\mathcal{X}$)	.290	.235	.130	.080
boolean8, K=180 of 232 classes, 5 seeds, chance 0.0056, *a different algebra*				
1: **composition never seen**	**.906**	**.851**	**.758**	**.676**
95% CI over classes	[.88, .93]	[.82, .88]	[.72, .79]	[.64, .71]
2: depth-2 composites in library	.919	.847	.771	.699
3: depth-3 composites in library	.889	.842	.790	**.727**
4: depth-4 composites in library	.882	.806	.751	.706
cost of never composing, paired	—	−.004	+.032	+.030
Alternate order, same multiset	—	.876	.799	.691
Untrained encoder ($\mathcal{X}$)	.268	.197	.133	.077
Strongest non-learned	.333	.222	.122	.089†
S3 maximum	.233	.222	.122	.078
ϕ_∞, *complete* fingerprint ($\mathcal{X}$)	.072	.122	.061	.044

†the one cell in either corpus where the strongest control below axis C is *not* a S3 member: it is the S1 variable bag, at 16× chance. See below.

Three things replicate on both corpora. *(i) The $\mathcal{F}_3$ audit passes at every depth.* The trained encoder's class-level interval clears the whole $\mathcal{F}_3$ supremum's, disjointly, in all eight corpus-by-depth cells. **The tightest cell on each corpus is tighter than the family itself, because the bar quoted there is not a declared member**: on `oneVarPoly13` at d=1 it is $[.990, .999]$ against the *tree-local* bag's $[.755, .860]$, which sits above $\sup \mathcal{F}_3$'s .805 $[.750, .860]$ (attained by $\mathrm{WL}_{h=3}$), as it does at d=3 and d=4 too; and on `boolean8` at d=4 it is $[.639, .711]$ against the *variable* bag's $[.050, .133]$, an S1 control that outscores the whole of $\mathcal{F}_3$ there (.089 against $\sup \mathcal{F}_3$'s .078, attained by $\phi_{d=2}$) and is reported as a finding about that corpus. Definition 1 enumerates neither, so neither sets the supremum we claim against; we quote them because each is the *higher* bar at its cell. *(ii) Order is not the fragile axis, anywhere.* The length-identical alternate order sits within noise of the forward chain on `oneVarPoly13` (+.001, −.005, −.004, every interval containing zero) and *above* it on `boolean8` (+.024, +.041, +.016), and all 24 orders are represented in both. *(iii) Proposition 5 holds again.* ϕ_∞ scores below the S3 maximum on both corpora at every depth (.290 against .810 on `oneVarPoly13` at d=1, .072 against .233 on `boolean8`) and decays toward chance with depth, so the supremum in Definition 1 is attained at coarse members on three corpora and two algebras rather than on one draw. **One tie, disclosed because the verifier pins it either way**: against the *saturated* member $\phi_{d=8}$ rather than against the supremum, ϕ_∞ is strictly lower on `boolean8` at all four depths but merely *equal* on `oneVarPoly13` from d=3 on (.130, .080). That is a fact about a single-variable corpus: argument order carries almost nothing extra to be penalised for, and it leaves the claim above untouched, since that claim compares ϕ_∞ to the family *supremum*, which is strict in all eight cells.

One thing does not replicate, and it is the mechanism claim. The clean coverage closure is polynomial-specific. On `poly8` and `oneVarPoly13` the test-depth-4 column rises monotonically in training depth (.918 → .955 → .980 → .987; .849 → .936 → .973 → .979) with every above-diagonal cell holding at ≥.988, which is what licenses reading the depth-4 deficit as a *coverage* deficit closed by seeing composites. On `boolean8` neither half of that holds. The test-depth-4 column reads .676 → .699 → **.727** → .706: it **peaks at training depth 3 and falls**, so the best any library does is recover .051 of a .230 drop, and the paired diagonal cost of never composing is only

$+.030$ $[.002, .058]$ against `oneVarPoly13`'s $+.130$ $[.106, .154]$. And the above-diagonal cells *do not* hold, five of the six sit *below* the never-composed arm at their own test depth, the depth-2 library being the only one that escapes: training on depth-3 composites reads .889 at test depth 1 against the never-composed arm's .906, and depth-4 training reads .806 at test depth 2 against .851, putting composites in the library *costs* shallow accuracy here. So on a different algebra the depth-4 gap is not mostly a coverage gap, and we do not know from these runs whether the residue is a capacity limit, interference between the four boolean rewrites, or a property of De Morgan in particular, which restructures the tree rather than padding it.

The coverage *price* grows with algebraic distance, and the margin grows with it. Over the never-composed arm the paired d=1 to d=4 drop is $+0.079$ $[.063, .097]$ on `poly8`, $+0.146$ $[.124, .169]$ on `oneVarPoly13` and $+0.230$ $[.197, .262]$ on `boolean8`, three times the published figure. Read alone that is the finding weakening. Read against the controls it is not, because the non-learned ladder falls faster still: the ratio of trained to strongest-non-learned at d=4 is $5.0\times$ on `poly8`, $2.3\times$ on `oneVarPoly13` and $7.6\times$ on `boolean8`. **`boolean8` therefore clears the criterion more decisively than `poly8` while paying three times as much for composition**, and `oneVarPoly13` clears it least decisively while paying half as much, the two orderings are not the same ordering, which is the argument for reporting the ratio and the drop separately rather than collapsing them into a single verdict per corpus.

An audit finding about `boolean8` in its own right. At d=4 its strongest control below axis C is the *S1* variable bag (.089 against chance .0056), not a S3 member (.078), and that bag reads .089 at d=1 too, unchanged with depth, as a bag invariant to every rewrite but $x \wedge \top$ must be. So roughly one in eleven of the 180 sampled anchors is identifiable from its variable multiset alone. This does not disturb the verdict (the encoder is $7.6\times$ above it), but it is a residual S1 cue of the kind §4.1's screen exists to surface, and it is *invisible* to that screen's corpus-level statistic, which is computed over EQNET's own class sets rather than over a 180-anchor sample and reports at most 3.2% across all 15 EQNET corpora. Sampling a corpus can reintroduce a cue the corpus does not have; we report the identity of the strongest control, not only its value, for that reason.

What these two runs do not bound. They are one architecture (Tree-LSTM; Appendix AK is where the architecture question is asked, and asked only of `poly8`), one recipe, one class partition per corpus and K at 200 and 180 rather than at each corpus's ceiling. Within-depth token count predicts identification slightly more strongly here than on `poly8`: the largest $|r|$ is 0.113 on `oneVarPoly13` and 0.187 on `boolean8` at d=4, against 0.088 on `poly8`, so size is a weaker null on the boolean corpus than the size-matched arm alone would suggest, small, and stated rather than omitted. The d=4 per-order spread is $[.725, .950]$ on `oneVarPoly13` and $[.480, .867]$ on `boolean8`, each over $\sim$7–8 classes per order, so the spread is noisy by construction and is printed to show that no single order carries the result rather than to support an inference about orders.

AN. Proofs of §3.2's Theorem and Propositions

Proof of Theorem 1. Write $s = \sup\{M(g) : g \in \mathcal{A}_P\}$, and let *chance* denote the score the protocol assigns a representation carrying no information about the label.

(i) Suppose $M(h) \leq s$, and take $g^\star \in \mathcal{A}_P$ with $M(g^\star) \geq M(h)$ (one exists: s is a supremum and $M(h) \leq s$, so either it is attained or some member exceeds any value strictly below it; if $M(h) = s$ is attained only in the limit, run the argument at $M(h) - \epsilon$ for arbitrary $\epsilon > 0$). Two regimes are then both realisable. In the *first*, h reads P and scores $M(h)$. In the *second*, h' computes $g^\star$ but is degraded by returning a fixed P-blind default on a measurable fraction λ of inputs; as λ runs from 0 to 1 the score moves from $M(g^\star) \geq M(h)$ down to chance, so some λ gives $M(h') = M(h)$, and h' is P-blind because $g^\star$ and the default both are. The two regimes therefore induce the same value of the score under audit. Moreover $M(g)$ is a function of g and the protocol alone: it does not depend on which regime produced the learned representation, so the observed pair $(M(h), M(g))$ is *identical* across the two regimes for *every* choice of g. No comparison of scores can separate them.

(ii) If instead $M(h) > s$, then no P-invariant representation attains $M(h)$, so the second regime of *(i)* is unrealisable and the observation is not attainable while discarding P. Combined with *(i)*, which

exhibits the ambiguity at *every* value $M(h) \leq s$, the condition $M(h) > s$ is therefore necessary as well as sufficient to remove it.

(iii) Let $g \notin \mathcal{A}_P$. Then $M(g)$ neither upper- nor lower-bounds s. It can exceed s: a comparator reading only the property at issue may outscore every P-invariant representation, which is what a variable bag does on AI Feynman (§1). It can fall below every member of $\mathcal{A}_P$: a P-sensitive comparator that is otherwise label-uninformative sits at chance, and $s >$ chance whenever any invariant control does better than chance. So no value of $M(g)$ decides the test in *(ii)*; and by *(i)* the pair $(M(h), M(g))$ is the same in both regimes whenever $M(h) \leq s$. Hence the comparison of $M(h)$ against an inadmissible $M(g)$ constrains neither regime, in either direction. □

What this theorem does and does not claim. It is a *necessity* result about comparator-based evidence, and it is deliberately weaker than it may read. *(a)* It licenses nothing: $M(h) > s$ rules out the P-blind regime and says nothing about *how* h computes what it computes, which is Proposition 3. *(b)* It bounds only evidence of this form. Mechanistic, interventional and behavioural evidence about P is not a comparison of protocol scores and is untouched. *(c)* **It does not give this paper more than the paper claims.** $\mathcal{A}_P$ is not enumerable, so what is actually run is a declared $\mathcal{F}_t \subseteq \mathcal{A}_P$, and $\sup \mathcal{F}_t$ is only a *lower* bound on s: exceeding it removes the ambiguity relative to that family and not absolutely. That is precisely why every claim here is **family-relative** at every rung where $\sup \mathcal{F}_t$ is a lower bound, the theorem says what the ideal instrument is and, by the same argument, that no *generally* realisable instrument is it. **One rung is the exception, and it is the one the paper's central result is read off**: at the shape-matched twin the twin *is* a realisable instrument computing s, because every $g \in \mathcal{A}_P$ ties there by construction, so $\sup \mathcal{F}_3 = s = 0.500$ exactly and the ambiguity is removed absolutely rather than relative to a family (Proposition 4). Elsewhere (UNSEENEQCLASS, the AI Feynman suites, every level below S3) the lower-bound reading above is the operative one, which is why Appendix BB searches those rungs adversarially instead of arguing about them. *(d)* It is not vacuous here. Clause *(iii)* has a measured instance in the paper's opening example: a zero-parameter variable bag scores 1.000 against a Tree-LSTM's 0.972 and refutes nothing about variable-identity invariance, because it reads nothing else. *(e)* **What is *not* definitional is the consequence §3.2 draws from the theorem: the statistic ranges over *trained* readouts, so every ceiling in this paper is a result rather than a stipulation** (Cor. 2). A reader who finds the theorem itself close to a restatement of what an admissible family means is not disagreeing with us, §3.2 presents it as scoping.

Proof of Corollary 1. $g \in \mathcal{A}_P$ and $M(g) \geq M(h)$, so $M(h) \leq s$ and Theorem 1*(i)* applies: the observation is induced equally by a P-reading and a P-blind h, so it is not evidence that h encodes P. □

The asymmetry noted after it in §3.2 is immediate: $M(g) < M(h)$ refutes the hypothesis that h is P-blind and says nothing about any other invariant g', which may still dominate h.

Proposition 2 (What passing the audit licenses)**.** *Passing the $\mathcal{F}$-relative admissibility audit through level ST rejects every St≤T explanation of $M(h)$ and licenses no claim beyond ST, and none about coverage at all, which only axis C addresses.*

Proof of Proposition 2. Each rejection is Corollary 1 at that level's g; the second clause is that asymmetry applied at $t{=}T$. □

Proposition 3 (No admissible family certifies)**.** *Let $\mathcal{F}_t$ be as in Definition 1 and let P be structure the cues at levels $\leq t$ do not determine. Membership requires invariance to P, so $\mathcal{F}_t$ contains no explanation of $M(h)$ that* reads *P; and by Proposition 1 it cannot be extended to contain one without forfeiting the invariance that made its members evidential in the first place. Hence $M(h) > \sup \mathcal{F}_t$ rejects exactly the explanations $\mathcal{F}_t$ encodes and licenses nothing about the rest, for* every *admissible family, so* ***certification is unreachable by admissibility rather than merely unbuilt by us****.*

Proof of Proposition 3. Let P be structure the cues at levels $\leq t$ do not determine, and let E_P be the explanations of $M(h)$ that read P, each realised by some representation that is not P-invariant. Definition 1 makes P-invariance a condition of membership, so $\mathcal{F}_t \cap E_P = \emptyset$; and Proposition 1 says any refinement reaching into E_P forfeits membership, so no admissible family contains a member

of E_P. The exclusion is not a gap that adjoining representations could close, because Corollary 1 applies only to a P-invariant g: a P-sensitive comparator could not falsify a P-claim even if it were admitted, however high it scored. So the audit's evidential reach is exactly $\mathcal{F}_t$, for any $\mathcal{F}_t$ meeting the definition, of any cardinality. □

What this proposition does and does not claim. It bounds *comparator-based* evidence of the form Definition 1 defines: a claim licensed by exceeding a family of invariant controls. It is *not* a claim that compositional reasoning is unmeasurable, nor that no evidence of it exists, mechanistic, interventional or behavioural evidence is simply not of this form and is untouched by the argument. Nor does it excuse a small family: the reach of an audit is exactly what it enumerates, so *which* controls are declared still decides how much a result rules out, which is why §4.2 stress-tests $\mathcal{F}_3$ rather than resting on Proposition 3. What the proposition does settle is the question a reviewer is entitled to press: enlarging the family is not a route to certification, so an audit's honest output is a falsification certificate naming its family, and reporting one is not a concession but the correct form of the result.

Proposition 4 (At the shape-matched twin the ceiling is exact, not family-relative)**.** *Let the protocol be the shape-matched twin of §4.2: each trial presents a held-out form T and a partner T' obtained from T by a sibling swap, and scores a representation g by whether g places T closer than T' to its class's centroid, a tie counting $\frac{1}{2}$. Let P be the arrangement the swap alters and $\mathcal{A}_P$ every representation invariant to it. Then $M(g) = 0.500$ for* every $g \in \mathcal{A}_P$*, so $s = \sup\{M(g) : g \in \mathcal{A}_P\} = 0.500$ and is* attained *rather than approached; and $f \circ g$ inherits the value for every readout f, at any capacity. Hence at this rung $\sup \mathcal{F}_3 = s$: the declared family is not a stand-in for the unenumerable $\mathcal{A}_P$ but computes its supremum* exactly*, and $M(h) > 0.500$ meets Theorem 1*(ii) *with no family relativisation to carry.*

Proof of Proposition 4. By construction the swap permutes two siblings and changes nothing else: T and T' carry the identical variable multiset, the identical operator/arity multiset and the identical bounded local shape, and differ in P alone. Let $g \in \mathcal{A}_P$. Invariance gives $g(T) = g(T')$, so for any centroid c (however computed, from whatever training forms, under any score reading only the representation) the two similarities are the *same number*, and the trial ties. Every trial ties, so $M(g) = \frac{1}{2}$ exactly, per instance and hence under any aggregation of instances; no member scores above it and none below, so the supremum is 0.500 and every member attains it. Post-composition preserves this: $f \circ g$ is P-invariant whenever g is, which is Proposition 1's argument applied to $\mathcal{A}_P$ in place of $\mathcal{F}_t$, so fitting a readout at any capacity cannot break the tie. Finally $\mathcal{F}_3 \subseteq \mathcal{A}_P$ gives $\sup \mathcal{F}_3 \leq s$, and the members of $\mathcal{F}_3$ attain 0.500, so that inequality is an equality. □

Two directions, one caveat, and they are not the same statement. Appendix AB records that a baseline *observed* at exactly 0.500 on every one of the 348 classes is admitted to $\mathcal{F}_3$ under an operational definition, and that this is a membership test rather than a proof of global invariance to S3. That caveat governs the direction *observation* ⇒ *invariance* and stands unchanged. Proposition 4 runs the other way, *invariance* ⇒ 0.500, which is a deduction from the hypothesis and needs no test to license it. Reading the proposition as weakening the caveat is exactly the conflation it is stated to remove.

What this proposition does and does not claim. *(a) P is the arrangement difference a given twin realises, not the whole of S3.* The exactness is exactness for that perturbation, and says nothing about rearrangements no twin was built for. **So the residual in this paper's central result is not *which representation* (every invariant one is covered, by proof) but *which perturbation class* a twin realises**, and that residual is measured rather than asserted: Appendix AA's three *nested* probes show that one rung looser than the twin the body reports, the test admits a control scoring 0.909 on the very perturbation membership requires it not to see, while at the strictly most-matched rung all three bags are pinned and the encoder gap stays positive; and on the rotation twin the same bags reach 0.889–0.901 (Table 25), which is why rotation is reported as the weaker probe. That is a narrower limitation than family-relativity, and a measured one. *(b) It rests on the label-blind tie rule*, which is implemented, disclosed and separately audited (Appendix AI); under a rule breaking ties by presentation order the value would be that rule's artefact rather than g's score, which is why the audit exists. *(c) M reads only the representation.* A procedure exploiting the order in which the pair is presented is not a representation of the form, so it is outside $\mathcal{A}_P$ by construction rather than

by exclusion. *(d) It licenses nothing about mechanism.* Removing the ambiguity is not certification: Proposition 3 still holds here, for every admissible family, so passing rejects the P-blind regime and says nothing about *how* h computes what it computes. **The proposition's content is that the incompleteness objection has no force at this one rung, not that the encoder is shown to compose.**

Proposition 5 (Resolution–retrieval tradeoff for complete feature representations)**.** *Under a protocol scoring g by similarity of feature supports, if g is* complete *(its features determine the form up to variable renaming), then $M(g)$ tends to the class-prior rate as held-out forms grow syntactically distant from their class's training forms. The supremum in Definition 1 is therefore attained at* coarse *members of $\mathcal{F}_t$, and adjoining a more resolving g cannot raise it.*

Proof of Proposition 5. A complete g's features are in bijection with the form's substructures, so two forms share a feature only where they share that substructure entire; a rewrite chain that rebuilds the root leaves none, the similarities collapse to a common 0, and the ranking is label-blind. Coarsening is what places equivalent forms together. □

What this proposition does and does not claim. It is a statement about the class of *feature-support similarity* protocols defined in §3.2, a g whose score is a similarity between feature supports, evaluated by retrieval against a class's training forms, and it inherits every assumption of that construction: the similarity, the (uniform) feature weighting, the completeness of g, and "syntactically distant" meaning a rewrite chain that rebuilds the root. It is *not* an impossibility theorem about structural representations, about retrieval metrics, or about learned encoders, none of which need be feature-support similarities. Its role in the paper is to *explain* a measurement, not to license one: the non-monotonicity below is measured directly, and would stand as a finding about $\mathcal{F}_3$ if the proposition were dropped.

Which step is protocol-specific, and which is not. The proof has two steps and only the first is tied to this protocol. *Step 1* (a complete g's feature supports become disjoint once a rewrite chain rebuilds the root) depends on the feature-support construction: a different similarity, or a weighted one, could soften it, and we claim nothing about how far. *Step 2* (once the supports are disjoint the ranking is label-blind, so $M(g)$ falls to the class-prior rate) is general, holding for *any* scoring rule that reads only the overlap of two representations, whatever built them. The consequence Definition 1 actually uses is weaker still and needs neither step: a supremum over a family whose members are *required* to be invariant to the structure their level does not determine cannot be raised by adjoining resolution, because resolution is what forfeits membership. That much follows from the invariance requirement alone, which is why the three-way distinction in §3.2 is stated there in terms of invariance rather than in terms of this proposition.

Corollary 2 follows by combining the two: Definition 1 quantifies over $\sup\{M(g) : g \in \mathcal{F}_3\}$, and by Proposition 5 that supremum is attained at coarse members, so no single member, in particular not the most resolving one, settles passing at S3. §4.2 measures the non-monotonicity on three corpora, two algebras and both protocols.

AO. Does the Twin and Unseen-Class Design Replicate on a Different Algebra? (tags `r85_boolean_swap_twin`, `r86_boolean_unseen_ladder`, `r86b_boolean_arch`, `r87_boolean_tier3`)

Appendix AM replicated the *composition* design off `poly8`; the two results §4.2 leads with, the shape-matched twin and the unseen-class ladder, still rested on one corpus and one algebra. This appendix runs both on BOOLEAN8, changing only `--corpus`, with `poly8` still the default so no shipped number can move.

A correctness note, stated because it would have produced a wrong number rather than a crash. BOOLEAN8 emits token ids up to 38; `VOCAB_SIZE` is frozen at 34 (Appendix AM), and it is `build_encoder`'s default. The *trained* encoder threaded a `vocab_size`; the *untrained control did not*. On MPS an under-sized `nn.Embedding` does not raise: it returns garbage, for the Tree-LSTM, the GIN and the Transformer alike, so the run would have reported a plausible trained–untrained gap computed against a broken control. The threading was fixed in both runners, every other `build_encoder` call site in the artifact was audited (the shipped boolean composition run

builds no random encoder; the probe matrix returns before building one on boolean corpora; `r67` maps boolean operators onto the arithmetic vocabulary deliberately), and the fix was verified two ways, neither of them a re-run of a trained row, Appendix AL reports those as not bit-reproducible across environments in the first place. *By construction:* on every arithmetic corpus the threaded value *is* `build_encoder`'s former default, so the model built on the `poly8` path is bit-for-bit the one the shipped logs were made with. *By re-run, on the part that is deterministic:* `r71`'s encoder-free ladder, re-run on the `poly8` default after the change, agrees with its shipped log on all 693 scalar cells with *zero* drift. The same shape, a control not reconfigured along with the model, is what produced the `r79` defect disclosed in Appendix AK.

Three ways BOOLEAN8 is not `poly8`, all of which make its numbers read differently. *(i) The twin pool is thinner.* Its only non-commutative binary operator is `implies`, and $73/190$ classes (38%) admit a swap twin passing all four acceptance tests, against $143/200$ (72%) on `poly8` at matched K. *(ii) Classes are larger, which makes the k-NN task easier in a way a ratio to chance hides.* BOOLEAN8 holds exactly 20 forms per class where `poly8`'s median is 6, so the unseen pool is 10, 20 and 38 classes at $K{=}50, 100, 190$, and its null is the exact random-label expectation, 0.09548, 0.04762 and 0.02503, *below* what $1/n_{\text{classes}}$ would report (0.1, 0.05, 0.02632); we quote the former. *(iii) K is capped at* 212, not at `poly8`'s 1000, by the same split arithmetic: 232 classes, of which K train and $\min(K/5, 100)$ are held unseen. Every boolean scale is therefore *below* the $K{=}200$ threshold at which `poly8`'s margin separates cleanly, which is the honest reading of any comparison here.

And a fourth, which is an audit finding about the auditor. BOOLEAN8's coarse bags *tie massively*, 20 same-inventory forms per class over a small vocabulary, so `r71`'s tie-quantisation guard fires where it never fires on `poly8`: {token n-gram, tree-local, WL_2, WL_3} are flagged unsafe at $K{=}190$ (24, 18, 1 and 3 ambiguous gaps against a 23-way tied k-th neighbour), against *zero* flagged representations at every `poly8` scale. We therefore read the $\mathcal{F}_3$ supremum off a tie-*safe* member at every scale, which costs nothing because the flagged rows score below it anyway, and the verifier pins the flagged set *as a set* so that a rerun moving the flag onto the member we quote fails rather than passes.

BOOLEAN8, UNSEENEQCLASS k-NN precision at $k{=}5$ (tag `r87`)

Representation	$K{=}50$	$K{=}100$	$K{=}190$
Exact random-label null	.0955	.0476	.0250
Variable-only bag (S1)	.287	.209	.113
Variable-blind bag (S2)	.368	.361	.238
Full-token bag	.535	.662	.417
Token n-gram bag‡	.675	.450	.429
Tree-local bag‡ ($\notin \mathcal{F}_3$)	.739	.518	.498
$\mathrm{WL}_{h=1}$ ($\in \mathcal{F}_3$)	.714	**.525**	**.511**
$\mathrm{WL}_{h=2}$, $\mathrm{WL}_{h=3}$ ($\in \mathcal{F}_3$)	.702, .692‡	.508, .500	.492‡, .483‡
$\phi_{d=1}$ ($\in \mathcal{F}_3$)	**.732**	.514	.494
$\phi_{d=2}$, $\phi_{d=4}$, $\phi_{d=8}$ ($\in \mathcal{F}_3$)	.681	.483	.478
ϕ_∞, *complete* fingerprint ($\mathcal{X}$)	.596	.401	.395

‡tie quantisation flagged unsafe at this scale; the supremum is read off a flagged-free member.

Proposition 5, measured a third time and in a second protocol. ϕ_∞ scores below *every* ϕ_d member at all three scales, $\phi_{d\geq 2}$ collapses to one value (saturation, verified as bag equality) and the $\mathcal{F}_3$ supremum is attained at a *coarse* member ($d{=}1$ at $K{=}50$, $h{=}1$ at $K{=}100$ and 190), never at the most resolving one. Corollary 2 therefore holds on two algebras and under both the composition and the unseen-class protocol.

The shape-matched twin replicates, and the gap *widens*. Over the 73 classes admitting an `implies` sibling swap, all four acceptance tests passing per instance, with non-equivalence decided by the exact 2^3-row truth table rather than `poly8`'s 24-sample numeric probe:

BOOLEAN8 shape-matched swap twin, K=190, 73/190 classes (38%), 5 seeds, chance 0.500 (tag r85)			
Encoder	trained	untrained	gap
Tree-LSTM	**0.934** ±.015	0.636 ±.052	**+0.299**
Transformer	0.606 ±.061	0.492 ±.103	+0.114
GIN	0.511 ±.026	0.477 ±.028	+0.034

Non-learned, same twins: full-token bag 0.500, tree-local bag 0.500 (both order-blind, hence pinned by construction); token n-gram 0.706 (order-*sensitive*, hence $\mathcal{X}$, not admissible).
poly8 *at matched* K=200*, 3 seeds,* 143/200 *classes:* Tree-LSTM 1.000, untrained 0.835, gap +0.166.

The boolean twin is the *harder* instance and the *cleaner* result. The trained encoder no longer saturates, so the measurement has headroom in both directions, while the untrained control falls much further (0.636 against 0.835), and the gap therefore *widens* to **+0.299** from +0.166. It is a per-seed separation, not a separation of means: the trained encoder's *worst* seed (0.918) sits above the untrained encoder's *best* (0.685). Both $\mathcal{F}_3$ bags are pinned at exactly 0.500 here as on poly8, and the token n-gram (the one non-learned control that *can* see a sibling swap, and an $\mathcal{X}$ stress test rather than a family member for exactly that reason) reaches 0.706, close to poly8's 0.664 and far below the encoder. **The curve is again this architecture's:** GIN sits at chance and the Transformer clears chance by 0.106 over a control that is itself at chance, so the twin ordering is Tree-LSTM $\gg$ Transformer $>$ GIN, the ordering Appendix AK finds on poly8 under a different design. One disclosure in the other direction: the shipped poly8 twin used 3 seeds and this run uses 5, so the boolean spread is estimated from more draws, not fewer. And the statistic that disclosure asks for, since one class *is* one twin: a rerun capturing both arms per class (r106_boolean_swap_twin_classci, the same 5 seeds, reproducing the untrained 0.636 and the Transformer's 0.492 to three decimals) bootstraps the paired per-class gap over all 73 twins as +0.296 $[+0.225, +0.364]$ for the Tree-LSTM, +0.119 $[-0.004, +0.238]$ for the Transformer and −0.004 $[-0.047, +0.040]$ for GIN. **Only the Tree-LSTM's gap interval excludes zero**, which sharpens the reading above rather than softening it: the per-arm seed spreads (±.026, ±.028) never gave GIN's +0.034 a sign, and the rerun's estimate falls on the other side of zero while staying inside an interval that contains it. GIN is also the only encoder here whose arms do not reproduce to three decimals (0.494 and 0.499 against 0.511 and 0.477, where both Tree-LSTM and Transformer *untrained* arms reproduce exactly), which is the same instability read off the runs rather than off the classes. Against poly8 at the matched K=200, whose +0.166 the same rerun reproduces with $[+0.119, +0.215]$, the boolean interval is *disjoint*, so the widening is not an artifact of comparing point estimates.

The unseen-class ladder, replicated across three scales and three architectures. Read against the same partition the previous table's supremum is taken over, r70 and r71 share the shuffle seed, the usable-class filter and the $\min(K/5, 100)$ unseen slice, and the two logs are asserted to agree on the pool and its exact null at every scale, since otherwise the comparison would not be legitimate:

BOOLEAN8, UNSEENEQCLASS k-NN precision at k=5, class-level 95% intervals (tags r86, r86b)			
	K=50	K=100	K=190
Exact random-label null	.0955	.0476	.0250
Strongest non-learned, *upper* interval bound	.855	.747	.596
Trained Tree-LSTM 5 seeds	**.940**	**.974**	**.955**
95% CI over classes	[.871, .989]	[.957, .987]	[.925, .979]
Untrained Tree-LSTM ($\mathcal{X}$)	.813	.769	.698
trained − untrained	+.127	+.205	**+.258**
Trained GIN 3 seeds	—	—	.881 [.850, .909]
Trained Transformer 3 seeds	—	—	.669 [.588, .746]
Untrained GIN, untrained Transformer	—	—	.713, .463

What replicates. Passing the $\mathcal{F}$-relative admissibility audit through S3 holds for the Tree-LSTM at *all three* scales, its class-level interval clears the upper bound of *every* non-learned interval in the previous table, where on poly8 the K=50 and K=100 intervals *overlap*, and the trained–untrained margin **grows with** K exactly as it does there (+.127 → +.205 → +.258). **What does not, and is reported for that reason.** *(i)* The *untrained* encoder exceeds the $\mathcal{F}_3$ supremum at every scale (.813 against .732, .769 against .525, .698 against .511), which on poly8 it does not: with 20 forms per class the unseen neighbourhood is easy enough that a random tree encoder beats every order-blind

fingerprint we can build. **Round-36 correction: this anomaly splits in two, and one half of it was our instrument rather than the family** (tag `r100`, BA). Those three suprema were each measured with *one* fixed unfitted nearest-centroid readout. Fitting the readout over the same order-blind maps raises the $K{=}50$ ceiling to **.838**, which **overtakes** the untrained encoder's .813: at that scale the gap was a *readout* artifact, and attributing it to the family was wrong. At $K{=}100$ and $K{=}190$ the opposite holds *provably*: .769 and .698 exceed .652 and .640, the supremum over **every** readout over **every** admissible map, so no member of the declared family can reach the untrained encoder there at any capacity. That half is a property of $\mathcal{F}_3$ on this corpus rather than of our choice of comparator, and we now state it that way. Either way the untrained encoder is billed as an $\mathcal{X}$ stress test rather than an $\mathcal{F}_3$ member (§3.2); it is not invariant to arrangement, so Corollary 1 does not apply to it, and here that distinction does visible work rather than none. *(ii)* SEENEQCLASS identification *falls* with K where unseen precision does not ($.952 \to .922 \to .911$), the two protocols having different nulls ($1/K$ against the class-size expectation); we quote both rather than the flattering one. *(iii)* The Transformer's interval does *not* clear the strongest non-learned bound ($[.588, .746]$ against .596), so it fails S3 here as it fails past $d{=}2$ on `poly8` (Appendix AK).

The one result that changes what the twin is for. GIN clears S3 on unseen classes and sits at *chance* on the twin, .881 $[.850, .909]$ against a strongest non-learned bound of .596, then .511 (sd .026) against a chance of .500. Nothing in the unseen-class protocol distinguishes a representation that composes from one that merely clusters well under a coarse similarity; the twin does, and this is the cleanest demonstration of it in the paper: *the same encoder, the same corpus, the same partition, one protocol passed and one at chance*. It is also the strongest form of §4.2's Finding 2, clearing the bags is not enough, because here the thing that clears the bags is a trained encoder rather than a strawman. A reader who accepts only the unseen-class number would have credited GIN with structural identification on this corpus.

AP. IS THE COVERAGE EFFECT SPECIFIC TO THE GENERATING SCHEMA? THE SCHEMA-TRANSFER MATRIX (TAG `R88_SCHEMA_TRANSFER`)

§4.5's intervention has two arms, and two arms cannot separate two explanations of the same gain. Including the generating schema may matter because *that* transformation was seen, or merely because *one more* out-of-library transformation was seen. This appendix crosses five training libraries with all four held-out sets, so the diagonal is the IN arm, every off-diagonal cell is a *wrong-schema* control, and the schema-free row is the OUT arm (tag `r88`, §4.5's recipe field for field, $5 \times 25 = 125$ encoders, 107.8 min).

What makes the cells comparable, and the one hazard the widening introduced. The population is the four schemas that apply to *all ten* equations, measured before the run rather than assumed, `reciprocal_mul` and `numerator_doubling` reach $7/10$, the two fraction splits $4/10$ and `const_reposition` $1/10$, so only these four give cells of equal K. Held-out trees are built *once*, outside the seed loop, so a column's test items are byte-identical across all five libraries; every library holds the same number of forms, so a contrast isolates *which* schema and never how much data. The hazard is the guard: §4.5 only had to keep a library's forced training form from reproducing *its own* held-out form, whereas here each encoder is scored against *four* held-out sets, so the forced form must avoid their union or a wrong-schema cell would train on its own test item. It was widened accordingly and the result is clean (0 collisions and 0 cells excluded for want of a valid forced form, over all 40 library-equation pairs) and the verifier recomputes all 40 held-out serialisations with an independent traversal and re-checks every forced form against them, rather than reading the runner's flag. **The schema-free row is not a new baseline**: it is §4.5's OUT arm on the same seeds through the same code, and its first ten seeds are asserted to equal the shipped `r59` log *exactly* in both arms, which anchors this matrix to the published experiment instead of inviting a comparison across runs.

	plain				renamed			
Training library	half	scale	triple	add	half	scale	triple	add
none (OUT arm)	.648	.296	.284	1.000	.376	.196	.176	.888
half_of_double	**.748**	.320	.304	1.000	**.412**	.200	.188	.880
scale_and_divide	.736	**.420**	.364	.992	.464	**.252**	.236	.848
triple_of_third	.748	.452	**.436**	1.000	.456	.264	**.248**	.852
add_subtract	.552	.260	.248	**1.000**	.280	.176	.168	**.892**

Columns are held-out schemas, rows training libraries; the diagonal (bold) is the IN arm. 25 seeds, $K{=}10$, chance 0.100.

The reading the run committed to before the prose existed. The runner decides between four pre-registered outcomes from its own pooled contrasts and writes the verdict into the log, which the verifier then pins, so a rerun landing elsewhere fails rather than leaves this appendix describing an outcome the log no longer shows. It returned `schema_specific`: pooled over the four columns, IN beats WRONG by $+0.070$ $[0.021, 0.115]$ plain and $+0.033$ $[0.019, 0.046]$ renamed, with each of the three columns that has headroom positive on its own interval ($+0.069$ $[0.036, 0.105]$, $+0.076$ $[0.029, 0.131]$, $+0.131$ $[0.064, 0.203]$). **The pooled figure is conservative**: `add_subtract` is identified perfectly in *every* library, so it contributes an exact zero to the mean rather than inflating it.

Two results that qualify it, both reported because they weaken a claim we would rather make. *(i) Out-of-library volume is not the mechanism, and can cost.* A wrong schema does help a little on average (WRONG minus OUT is $+0.024$ $[0.006, 0.041]$) but that effect **does not survive variable renaming** ($+0.009$ $[-0.016, 0.023]$, straddling zero) where the schema-specific effect does, and the `add_subtract` library sits *at or below* the schema-free row in **every** column (-0.096, -0.036, -0.036, 0.000). A library is not a bag of extra data. *(ii) The specificity is at the level of the transformation family.* `scale_and_divide` ($a \to 2a/2$) and `triple_of_third` ($a \to 3a/3$) are the same rewrite at a different constant, and that pair was pre-registered as a near/far contrast *before* the run for exactly this reason. Each transfers to the other almost completely: the near library scores 0.452 and 0.364 against far means of 0.290 and 0.276 which are themselves *below* the corresponding OUT arms (0.296, 0.284), so far transfer is nil, not merely small, and on the `scale` column the near library (0.452) in fact *exceeds* the generating schema's own 0.420. The renamed arm says the same thing more sharply: on the `half` column the diagonal is **not** the maximum, two same-family libraries reading 0.464 and 0.456 against the generating schema's 0.412.

What this changes and what it does not. §4.5's causal claim is unaffected and its replication is stronger, IN minus OUT pooled is $+0.094$ $[0.031, 0.139]$ here at 25 seeds against `r59`'s $+0.061$ $[0.019, 0.123]$ at 10, but the *mechanism* sentence has to be family-level, and §4.5 now says so. Two limits. The near/far grouping rests on **one** pre-registered pair, which is all four universal schemas afford; the wider reading that `half_of_double` belongs to the same constant-scaling family (its column is lifted by all three of them and depressed only by `add_subtract`) is *post hoc* and is not asserted as a finding. And "family" here is our own descriptive grouping of four hand-written schemas, not a formal equivalence on rewrite systems; Corollary 1 applies to it as it does to everything else in this paper, no finite matrix of this kind can establish what a representation would do on a family it was never shown.

AQ. What Prevents a Specialised Invariance to These Four Primitives? Leave-One-Primitive-Out (tag `r89_leave_one_primitive_out`)

§4.3's composition result trains on single applications of four rewrite primitives and tests on their unseen compositions, which invites one objection above all others: nothing in that design distinguishes an encoder that has learned *composition* from one that has learned a specialised invariance to those four transformations. This appendix answers it by the only means we know that is not rhetorical, transplanting §4.5's intervention from schemas onto the primitives themselves. For each held-out primitive h, two encoders are trained over *one* class set: the IN arm's library holds single applications of all four primitives, the LOPO arm's holds the other three backfilled to the same

k_{para}=6, and both are scored on a test ladder whose drawn order places h *first*, so every depth-d test form contains h and the held-out trees are byte-identical across the two arms. The contrast is which primitive the library contains, never how much data it holds ($5 \times 25 = 125$ encoders, 385.6 min).

The three hazards the restriction introduces, and what was done about each. *(i) The closure guard must keep all four primitives.* It rejects a candidate whose declared depth overstates the rewrites actually needed; searching only the library's three would let a depth-4 form reachable in fewer steps *via* h pass a guard that never looked for h. The guard therefore searches the full four under sample seeds independent of the construction's (5077 against 1013), and found 0 test forms inside the $(d-1)$-closure over all 16 ladder–depth cells. *(ii) Backfilling from three primitives raises the chance that a filler form coincides with a test form.* The verifier recomputes this rather than reading the runner's flag, intersecting every arm's library against the held-out forms of all four ladders at all four depths: 0 collisions over 200 classes $\times$ 5 arms $\times$ 4 ladders $\times$ 4 depths. *(iii) An unequal class set across arms would make the matrix unpaired.* 200 classes are retained in all five arms simultaneously, from 341 scanned, and `library_underfilled` dropped *none* of them. Beyond these, the verifier reimplements the parser, the serialiser and all four rewrites independently of the runner's, so that the held-out primitive's absence is established by enumerating *every* site of every anchor rather than the sites the runner happened to sample, and confirmed a second way by stripping each insertion primitive's own syntactic signature back out of its library (0 reductions); every one of the 9400 serialisations in the log round-trips byte-for-byte between the two implementations, and every held-out form's token count exceeds its anchor's by exactly the sum of the drawn primitives' deltas (3200/3200 triples), so size cannot be the cue.

		LOPO: that primitive held out				**cost of holding it out**, paired			
Held-out primitive	Δtok	d=1	d=2	d=3	d=4	d=1	d=2	d=3	d=4
`commute`	0	.548	.555	.545	.523	+.421	+.404	+.398	**+.390**
$X \to (X + 0)$	4	.434	.422	.407	.393	+.565	+.560	+.548	**+.516**
$X \to (X \cdot 1)$	4	.354	.357	.353	.346	+.646	+.617	+.592	**+.556**
`double_negate`	8	.283	.269	.256	.242	+.713	+.718	+.703	**+.678**
pooled	—	—	—	—	—	+.586	+.575	+.560	+.535
IN arm, over four ladders	—	.991	.975	.951	.911	*its replication target:*			
`r81` *never composed*	—	.997	.983	.950	.918	Table 29, row 1			

Rows are the held-out primitive, ordered by the token delta it inserts; columns are test depth. 25 seeds, K=200, chance 0.005. The IN arm reproduces Table 29's published never-composed row to within 0.008 at every depth, so the LOPO arms are measured against a replication of the paper's own experiment rather than against a fresh and possibly weaker control.

The reading the run committed to before the prose existed, and the answer to the objection. The runner chose among pre-registered outcomes from its own paired contrasts and wrote the verdict into the log, which the verifier pins. It returned `specialised_collapse`: every one of the four intervals excludes zero, pooled +0.535 [0.502, 0.568] at d=4, **an order of magnitude above the** +0.079 **cost of never having composed** (Appendix AK's reproducible Tree-LSTM row; the body prints the d=8 depth cost, +0.200, not this one). **So the objection is conceded, and measured: the invariance is specialised to the primitives the library covers.** But the collapse has a shape, and the shape is the more useful answer. At depth 1 the LOPO encoder sits *on* its untrained control, .548 against .550, .434 against .402, .354 against .399, .283 against .317, never more than 0.045 apart, so for a primitive the library never showed, training buys *nothing at all*. Yet across depths 1–4 the LOPO encoder is nearly *flat* (its worst decay is 0.041) while that same untrained control collapses over the identical ladder (decays of 0.384, 0.211, 0.215, 0.133), and every LOPO arm still clears the strongest non-learned baseline at d=4. **What training buys on this corpus is invariance to composition *depth*, which survives holding a primitive out; not identification *level*, which does not.** That is a sharper scope statement than §4.3 could make on its own, and it is Finding 3's coverage mechanism appearing a fourth time, now over primitives rather than schemas.

Both pre-registered predictions failed, and the second failed in a way worth stating precisely. §4.5's family result, transplanted, predicted two things, written into the runner before it

ran. **P1: holding out `commute` (structural, with no family member left in the library) costs most. Refuted, and reversed**: `commute` is the *cheapest* primitive to hold out ($+0.390$) and `double_negate` the costliest ($+0.678$), their intervals disjoint. **P2: the two identity insertions, each the other's family member, cost least. Refuted as a *tie***: the identity pair averages $+0.536$ against $+0.534$ for the other two, a separation of 0.002. The honest statement is not that the family reading inverts on primitives but that it has *no explanatory power* here, the variation is entirely within its groups, not between them. §4.5's family claim stands where it was measured, on schemas; it does not transplant, and we would rather report that than quietly rescope it.

What does order the four costs, offered as *post hoc*. The ordering is monotone in Δtok, the number of tokens the primitive inserts: $0 \to +0.390$, $4 \to +0.516$ and $+0.556$, $8 \to +0.678$, with the three token-delta groups' depth-4 intervals **pairwise disjoint** and the two primitives that share a delta *not* separated from each other ($[.484, .548]$ against $[.523, .589]$). That the within-group pair is indistinguishable while the between-group gaps are clean is what makes this an ordering by token delta rather than four unrelated numbers. One confound is ruled out and one is not. Ruled out: at $d{=}4$ the untrained control is flat across the four ladders (.166–.191, a spread of 0.025), so the ordering is not inherited from how hard the ladders are, at $d{=}1$ it is *not* flat (.317–.550), which is why the claim is made at depth 4 and not pooled. Not ruled out: Δtok is confounded with how much of the tree the primitive rewrites, and with four primitives no design we can run on `poly8` separates those. **This is a hypothesis the experiment generated, not one it tested**, and it is asserted in the verifier only as the inequality the log actually shows, so that a rerun ordering them differently fails rather than leaves this paragraph standing.

AR. The Ladder Against a Model We Did Not Train (tag `r90_external_audit_lample`)

Why this run exists. Every constructive result above is on a corpus whose equivalence classes we built, which bounds how far the *evidentiary* argument has been shown to travel. This run applies the ladder of §3.2 to a benchmark and a model we had no hand in: the published Lample & Charton (2020) integration encoder (79,866,880 parameters, 6 layers, $d{=}1024$, the released `fwd_bwd_ibp` checkpoint trained on millions of FWD+BWD+IBP problems), evaluated on 320 integrands parsed from their own test distribution and labelled by integration family (polyrational, radical, trig, explog; 80 each).

Protocol, and what makes the rows comparable. One protocol for every row: nearest-family-centroid over L2-normalised representations, 25% of each family held out, 25 seeds, chance 0.25. **The splits are generated once per seed and shared by every row**, so rows differ only in the representation. Similarities are computed in `float64` and quantised at `TIE_DECIMALS`$=6$, with ties broken by one fixed label-blind permutation of the family indices, not by `argmax`'s first-maximum rule, which on the S1 row would have reported 1.000 for one family instead of chance.

The unit of inference is the integrand, not the family. Definition 1 asks for class-level intervals, and with four families a class bootstrap is not available. The interval is therefore a cluster bootstrap over *held-out integrands*: each item's score is its mean correctness over the seeds in which it was held out, and items are resampled. This is a weaker unit than the class bootstrap used everywhere else in the paper and is reported as such.

The verdict, and the reading that is *not* available. The encoder clears S3 and both stress tests and is beaten at S2 by 0.070, so Definition 1 licenses no claim above S2 from its number. It would be wrong to read this as a defect in their benchmark. The four family labels are *defined* by which operators appear, so an operator bag is not exploiting a flaw; it is reading the label. What the framework contributes is the entitlement: this is a S2 task, correctly identified as one, and a number on it is evidence of operator-inventory sensitivity and of nothing above that. An earlier appendix of ours described the same phenomenon as a "label-definition confound" and treated it as a caveat on the audit; in level language it is not a confound but a verdict, which is the clearest illustration we have of what the vocabulary buys.

Table 31: **The level ladder on an external benchmark and an external model** (320 Lample & Charton (2020) integrands, 4 families, 25 seeds, chance 0.25; tag `r90`). Intervals are item bootstraps. The encoder row is read from the log; reproducing it needs the published ∼1 GB checkpoint (`REPRODUCE.md` R26), and every zero-parameter row is recomputed from the shipped registry by `verify_claims.py`.

Representation	Level	Accuracy	95% CI
Variable bag	$\mathcal{F}_1$	0.250	[.207, .301]
degenerate by construction, all decisions tied			
Operator/arity bag	$\mathcal{F}_2$	**0.941**	[.925, .969]
Token bag	$\mathcal{F}_2$	0.851	[.817, .886]
WL kernel h=1 *(the $\mathcal{F}_3$ supremum)*	$\mathcal{F}_3$	0.818	[.779, .856]
WL kernel h=2 / h=3	$\mathcal{F}_3$	0.793 / 0.776	
ϕ_d, d=1 / 2 / 4 / 8	$\mathcal{F}_3$	0.752 / 0.734 / 0.726 / 0.726	
ϕ_∞, complete fingerprint	$\mathcal{X}$	0.726	
Tree-edit distance NN	$\mathcal{X}$	0.812	[.777, .852]
Untrained Tree-LSTM	$\mathcal{X}$	0.782	[.753, .820]
Published 80M encoder	subject	**0.872**	[.847, .913]

The gap this run does *not* close, and it is the one worth naming. Both external audits, this one and the NeSymReS arm of Appendix I (tag `r62`), stop at **S2**. **No pretrained model has been run on the shape-matched twin**, which is the one rung whose ceiling is known by proof (Prop. 4) and the rung our own constructive claim rests on. So the evidence that the *method* travels to models we did not build is real but **level-bounded**: it covers notational and operator-inventory shortcuts on other people's benchmarks, and says nothing about whether a released encoder would clear a twin. Building twins for an integration benchmark needs an equivalence oracle for that benchmark, which is why this is stated as an open gap rather than deferred to a footnote, **it is the single largest piece of coverage this paper does not have**.

A note on spelling. §3.2 renames the representational levels S1–S3 and separates axis C from them. The *shipped logs and verifier keys predate that rename* and still spell them `tier1`, `tier2`, `tier3` (e.g. `tier2_operator_arity_bag`, `passes_tier3_audit`); they are data and were deliberately left untouched, so a key naming "tier t" is the level this paper calls St.

Two things replicate here that were only measured on our own corpora. $M(\phi_d)$ is non-increasing in d (0.752 → 0.726) and the WL family non-increasing in h (0.818 → 0.776), with the order-sensitive completion ϕ_∞ at the family's floor. So the $\mathcal{F}_3$ supremum is attained at the *coarsest* member on a corpus built by other people, with trees deeper than any of ours: Corollary 2 and Proposition 5 measured externally rather than argued. S1's degeneracy is likewise a construction check, not a finding: these are single-variable functions, so a variable bag cannot separate anything and *must* sit at chance, and the assertion that all 2000 of its decisions are ties is what proves the tie rule is doing the work rather than the label order.

AS. Is the Composition Result an Artifact of Local Rewrites? (tag `r91_nonlocal_composition`)

The objection this run exists to answer. Every primitive in §4.3's four-primitive set edits a bounded neighbourhood of one node. A reader can therefore ask whether the flat composition curve measures composition at all, or only that the encoder is invariant to *local* perturbations. The test has to be a *swap*, not an addition: replace exactly one primitive with a non-local one, hold the class set, the skeleton order, the library size and the form sizes fixed, and read the paired difference.

Why distributivity, and why it is the only candidate. We measured the options over all 1102 `poly8` anchors rather than choosing one: `_rw_distribute` ($a{\cdot}(b{+}c) \to a{\cdot}b + a{\cdot}c$) fires on 283, `_rw_reassoc` on **0** (EQNET forms are already right-nested) and the two removal schemas on 0 (an anchor has nothing to remove). Distributivity is non-local in the sense the objection needs: the rewritten subtree a appears *twice* in the output, so the edit is not confined to a bounded neighbour-

hood and the output size depends on the size of what was rewritten. **No rewrite schema was written for this run, `equivalence.py` is untouched, and with it every published number.**

What is held fixed, asserted rather than stated. Arm L is the published set (`commute`, `double_negate`, `add_identity`, `mul_identity`); arm N replaces the last with `distribute`. Distributivity's token delta is $+4$ exactly when the duplicated subtree is a leaf (262 of the 283 anchors), so **every application in this run is filtered to its $\Delta{+}4$ sites**: the delta multiset is $\{0, 8, 4, 4\}$ in *both* arms, the depth-d delta is the sum over the drawn primitives and cannot depend on their order, and the run asserts per class against the serialiser that the depth-4 delta is $+16$ for every order in both arms. Mean test-form tokens are identical arm to arm at every depth (17.42, 21.24, 25.54, 29.00), the depth-1 test form is byte-identical across arms and absent from every library in both, library size is 6 in every arm, and SymPy verified all 532 ladder forms per arm equivalent to their anchors (0 refuted, 0 unresolved). The closure guard runs at each depth as in Appendix AH: 0 test forms inside the searched closure of the library.

Two disclosures the corpus forces, both consequences of the swap. (i) $K{=}133$, not 200. Of 1102 anchors, 840 lack a site for some primitive, 121 cannot build the arm-N depth ladder at $\Delta{+}4$, and 8 fail the leader constraint. **(ii) 18 of the 24 orders.** `_rw_distribute` has two or more distinct sites on **0** of 1102 anchors, so one site cannot both fill the library and supply a held-out single application: the swapped primitive can never lead the permutation. **The same constraint is imposed on arm L**, so both arms run the same 18 orders over the same classes. Retained swap positions are not uniform (54/47/32 at positions 1/2/3), because a late position must survive the rewrites applied before it.

Table 32: **One primitive swapped local → non-local, over one paired class set** (`poly8`, $K{=}133$ classes, 18 of 24 orders, 5 seeds, chance 0.0075; tag `r91`). Ladder L is the published four-local set; ladder N replaces `mul_identity` with `distribute`. LOPO holds the swapped primitive out of the library entirely, one encoder per seed, scored on both ladders, so every difference of differences is paired per class. Rows below the rule are the non-learned ladder run on both arms. **The alternate-order row carries a correction made in this revision**: every row here is the *full* library, and that row previously averaged two libraries (see below).

	$d{=}1$		$d{=}2$		$d{=}3$		$d{=}4$	
row	L	N	L	N	L	N	L	N
composition never seen	.982	.987	.964	.973	.949	.947	**.922**	**.908**
depth-4 composites in library	.999	1.00	.987	.999	.981	.991	.977	.985
swapped primitive LOPO	.996	.996	.761	.875	.597	.782	.454	.704
alternate order, same multiset	—	—	.998	.998	.938	.962	.893	.929
untrained encoder ($\mathcal{X}$)	.459	.426	.414	.310	.242	.208	.174	.134
strongest non-learned (bag)	.218	.181	.353	.271	.271	.263	.173	.263
$\sup \mathcal{F}_3$ (attained at WL, $h{\leq}3$)	.030	.015	.038	.015	.038	.015	.030	.015
ϕ_∞, *complete* ($\mathcal{X}$)	.000	.000	.015	.008	.008	.000	.015	.000

The contrast the run was built for. N − L on identification is $+0.004$ $[-0.006, 0.018]$, $+0.009$ $[-0.014, 0.035]$, -0.002 $[-0.030, 0.029]$ and -0.013 $[-0.054, 0.024]$ at depths 1–4: **every interval contains zero**. The within-arm cost of never composing is $+0.056$ $[0.027, 0.087]$ for ladder L and $+0.077$ $[0.048, 0.110]$ for ladder N at $d{=}4$, ladder L's replicates Table 29's $+0.069$ on this smaller class set. **So the composition result is not locality-specific over what is tested**, and §4.3's named scope, *composition-of-known-transformations*, covers a primitive set containing a rewrite that duplicates a subtree. This is the branch pre-committed in the module docstring before the run; the sign was filled in afterwards and the wording was not.

What did move, and it is coverage. Holding the *non-local* primitive out of the library costs $+0.204$ $[0.161, 0.250]$ at $d{=}4$; holding the *local* one out costs $+0.468$ $[0.391, 0.540]$; the difference of differences is -0.263 $[-0.344, -0.184]$, excluding zero at every depth ≥ 2. Both LOPO rows come from the *same* encoder per seed, scored on two ladders, so this is a paired within-encoder contrast. **Coverage cost is therefore primitive-specific, and a structurally bigger rewrite is not**

a harder one, which is Finding 3's mechanism sharpened, not weakened: what a library must contain is a member of the held-out form's rewrite *family* (Appendix AP), and `distribute`'s family is evidently better covered by the three remaining primitives than `mul_identity`'s is. At d=1 both coverage costs are slightly *negative* (-0.013, -0.009): a three-primitive library trains on fewer, easier singles, which helps at depth 1 and cannot help above it.

The centerpiece replicates a sixth time, and here it is only an ordering. In all eight cells (four depths $\times$ two arms) the $\mathcal{F}_3$ supremum is attained at a *Weisfeiler–Lehman* member with $h \leq 3$, never at a ϕ_d and never at the order-sensitive completion, and ϕ_∞ sits *strictly* below both that supremum and the coarsest member, the tree-local bag: Corollary 2 again, now over a primitive family containing a non-local rewrite. **We report it as a sign, not an effect size.** Chance on this class set is $1/133 = 0.0075$ and the entire $\mathcal{F}_3$ ladder lives between 0.000 and 0.038, so these are near-chance orderings on a hard task; the load-bearing measurements of the non-monotonicity remain §4.2's `poly8` ladder, the five class partitions (Appendix AL) and the external audit, where the values are far from the floor.

The confound, and its direction. The token *multiset* is not matched: an arm-N form contains a duplicated subtree that no arm-L form does. **A bag sees that, so the cue raises the non-learned rows rather than hiding beneath them**, the strongest non-learned row is 0.263 in arm N against 0.173 in arm L at d=4, while the untrained encoder moves the *other* way (0.134 against 0.174). The within-depth correlation between test-form size and being identified is logged for both ladders and is small and positive throughout (r=0.18–0.24 at $d\leq$3; undefined at d=4, where every form has exactly 29 tokens). Read conservatively, the size cue inflates arm N's LOPO row, so the coverage difference above is an *upper* bound on how much cheaper the non-local rewrite is to do without; the primary contrast, N $-$ L on identification, is unaffected because both arms' trained rows are equally exposed to it.

A correction to this appendix, and it is the largest we have had to make. Earlier revisions reported this run's alternate-order arm (a different permutation of the *same* depth-d multiset, outside the library closure) as costing ladder L $0.922 \rightarrow 0.671$ at d=4, "$+0.250$ paired", flagged as a non-replication of `r81`'s free order and read as evidence that a second order of the same multiset is materially harder on this class set. **That number was an artifact of our own aggregation, and no order effect of any size survives its removal.** The runner accumulates alternate-order scores into a per-*ladder* list from inside a loop over training *libraries*, so the alternate-order cell averaged the full library together with the leave-one-primitive-out library, while the forward cell it was subtracted from was the full library alone. The reported quantity was thus a contrast between two *different training libraries*, not between two orders. **The signature is exact rather than argued**: in all six cells the published delta equals the full-library forward score minus the two-library alternate-order mean (d=4 L: $0.9218-0.6714 = +0.2504$), and the ten per-seed values split cleanly into five strong and five weak ones by library.

Corrected, and the sign is no longer even consistent. Read within each library the order cost at d=4 is $+0.029$ on ladder L and -0.021 on ladder N; across all twelve library-by-ladder cells at $d\geq$2 no cost exceeds 0.035 in magnitude and *six of the twelve, exactly half, are negative*, so **all six of the run's stored "interval excludes zero" flags are spurious** and the corrected table row above is at or above the forward row at d=2. `r94`, a separate run built to test the reading this flag motivated, reproduces the corrected ladder-L cost to four decimals from an independent RNG stream ($+0.0286$; Appendix AV), which is how the defect was found. **So §4.3's order claim is not narrowed by this run after all**: the apparent counterexample dissolves, and Figure 3's arrangement bar is now *corroborated* on this class set and site filter rather than contradicted by them. **We record it as the fifth correction to a claim of our own, and note what kind of error it was**: a cost that appears only when two unmatched training conditions are pooled is precisely the failure this paper exists to name, and we did not catch it until we ran the experiment the appendix admitted was missing. The verifier now recomputes the decomposition and pins every within-library cost, so the artifact cannot reappear silently.

Reproduction. `python3 run_r91_nonlocal_composition.py --K 140 --num_seeds 5 --verify_classes 200` (≈44 min on one Apple M-series GPU;

REPRODUCE.md R27). Three trainings per seed, not four: the two LOPO rows are one encoder scored on two ladders, which is also what makes their difference paired.

AT. IS THE $\mathcal{F}_3$ VERDICT A PROPERTY OF THE FAMILY WE CHOSE? (TAG R92_FAMILY_STRESS)

The objection this run exists to answer. $\mathcal{F}_3$ is enumerated by us, so a reader can grant every number in this paper and still say that the encoder beats *our* family rather than the space of non-compositional explanations. The objection cannot be answered by argument, only by widening the family with descriptors chosen by someone else and re-reading the supremum. A reviewer supplied the list: subtree kernels, spectral graph features, graphlet counts, AST kernels, path kernels, tree automata, compressed subtree hashes.

Three of the seven are already in the ladder, under other names. The unordered subtree kernel and the bottom-up tree automaton are both the *same object* as bag_canon_blind, the ϕ_d family's own limit (Appendix AC); compressed subtree hashing is precisely WL's relabelling step, which is how WL_h is computed here; and an AST kernel over unordered children is ϕ_d. Saying so is part of the answer rather than a dodge, the family was assembled from the canonical order-blind constructions, not sampled from them. The four genuinely new families, five members in all, are **higher-order WL** ($h{=}4, 5$), a **graphlet census** (all connected 3- and 4-node subsets by canonical shape, not a re-parameterisation of ϕ_d, which always takes *every* child of an included node), a **path kernel** (multiset of root-to-leaf label paths) and a **Laplacian spectrum** (quantised eigenvalues of the unlabelled tree).

Admissibility is measured, not assumed, and the test has a control that must fail. Membership requires invariance to the structure the level's cues do not determine, and §3.2's operational test is the shape-matched swap twin: a member scores *exactly* 0.500 on *every* instance. Run on the paper's own $K{=}500$ swap split (348 classes), all five candidates are pinned at 0.500 per instance and are admitted. The two controls behave as they must: bag_tree_local, order-blind and therefore admissible, is pinned at 0.500, and the order-sensitive token n-gram reaches 0.681 and is therefore *not* admitted, without that second control the test could not distinguish a member from an extension, and Part A would establish nothing.

The widened family, UNSEENEQCLASS k-NN precision ($k{=}5$; tag r92, same split, null and tie rule as r71/r87)

	poly8					boolean8		
K	50	100	200	500	1000	50	100	190
Tree-local bag (admitted, *not* declared)	.726	.649	.493	.277	.277	.739	.518	.498
$\mathrm{WL}_{h=1}$ ($\in \mathcal{F}_3$, published)	.729	.641	.488	**.276**	**.278**	.714	.525	.511
sup over the *declared* $\mathcal{F}_3$	.736	.641	.488	**.276**	.278	.732	.525	.511
$\mathrm{WL}_{h=4}$ *(new)*	.726	.638	.486	.276	.277	.688	.491	.478
$\mathrm{WL}_{h=5}$ *(new)*	.726	.638	.486	.276	.277	.688	.489	.477
Graphlet census, 3–4 nodes *(new)*	.736	.641	.488	.276	.277	.728	.538	.501
Path kernel, root-to-leaf *(new)*	.732	**.649**	**.497**	.275	.276	**.749**	**.563**	**.512**
Laplacian spectrum *(new)*	.197	.137	.045	.020	.020	.333	.159	.087
sup over *declared* + five *new*	.736	.649	.497	**.276**	.278	.749	.563	.512
Trained Tree-LSTM, class-level *lower* bd.	.613	.760	.801	**.857**	.820	.871	.957	.925

What moved, and what did not. The supremum moves at 5 of the 8 cells and always to the *path kernel*, by at most $+\mathbf{0.038}$ (boolean8 $K{=}100$, $.525 \to .563$) and by $+0.009$ on poly8 ($K{=}200$, $.488 \to .497$). **No verdict moves anywhere**: the widened supremum stays below the trained encoder's class-level lower bound at every scale the paper claims, on both algebras. At poly8's $K{=}500$, the cell §4.2 prints, the supremum does not move at all: $\sup \mathcal{F}_3$ remains 0.276 and *every* new member lands at or below it, so the sentence *"sweeping the family adds no stronger member"* survives the widening verbatim. **One row here is above the declared family and is reported rather than hidden**: the tree-local bag reaches 0.277 at that cell, and 0.649/0.493 at $K{=}100$/200. It is *admitted* (it is pinned at 0.500 by the twin test like every other row), but Definition 1 does not enumerate it, so it does not set the $\sup \mathcal{F}_3$ the paper quotes; adding it to the catalogue raises the widened supremum in exactly one cell, $K{=}500$, from 0.276 to 0.277, against a trained lower bound of 0.857. That is the whole of its effect, and it is the kind of member this widening exists to price in. poly8 $K{=}50$ is the one cell where the family exceeds the encoder's lower bound, and it does so

with the *published* family too: that is the disclosed small-K overlap, which is why the paper claims a passing $\mathcal{F}_3$ audit only from $K \geq 200$.

Two things come free, and both cut against the intuition the objection rests on. *(i)* WL_h is **non-increasing in h in all eight cells**, strictly in seven, adding rounds of neighbourhood aggregation makes the control *weaker*, so Proposition 5 now replicates on a resolution axis we did not choose either. *(ii)* The Laplacian spectrum, the most mathematically sophisticated member here, is the *weakest* control in the family by an order of magnitude (.020 at $K{=}500$ against tree-local's .277). Neither observation changes a verdict, and that is the point: **admitting more members cannot lower the supremum, and here it barely raises it**, what an audit must quantify over is the family, and its *size* is not the statistic.

Two disclosures. The new WL members compress labels with `blake2s` rather than the salted `hash()` the published rows use (Appendix AI); re-running $h \leq 3$ under both hashings agrees to the fourth decimal at every cell, so the change is bookkeeping. And the Laplacian rows are quantised to three decimals to form a histogram key, which can only *merge* distinct spectra and therefore cannot inflate that row.

Reproduction. `python3 run_r92_family_stress.py` (≈72 s, CPU; no encoder is trained and the trained rows are read from the stored `r70`/`r86` logs; `REPRODUCE.md` R28). The published baggers are re-run alongside the new ones in the same call and reproduce `r71`/`r87` to four decimals, which is the cross-run determinism check.

And the exact version of the question, over every sub-family (tag `r98_family_robustness`). Widening the family answers "would other descriptors have changed the verdict?" only for the one wider family we happened to build. The general question, *how much could our choice of family have mattered at all?*, is exact rather than empirical, because the audit statistic is a **supremum** and therefore monotone under inclusion (Corollary 3). `r98` re-reads this run's log, never re-running it, and enumerates *all* $2^{13} - 1 = 8191$ non-empty sub-families of the 13-descriptor catalogue, Definition 1's seven declared members, the tree-local bag, and the five new ones, in each of the 8 cells. **Of 65,528 (sub-family, cell) pairs, 65,527 return the union's verdict** (0.999985), and **every one of the 7 cells the union passes is passed by all 8191 of its sub-families**, monotonicity verified by enumeration and not only by algebra. The run additionally asserts the closed form $2^{n-a} - 1$ for the number of passing sub-families given a blocking descriptors, and cross-checks each reconstructed supremum *and the member attaining it* against this appendix's own recorded values.

The single flip, named. It is at `poly8` $K{=}50$, the one cell the union already reports as a **failure**, and the flipping sub-family is {`bag_laplacian_spectrum`} alone, which scores 0.1968 there where the other twelve descriptors score 0.7258–0.7355 against a trained lower bound of 0.6126. So the flip is a *narrower* family failing to refute what the union refutes: it makes the audit weaker, in the direction the supremum rule already requires. **No sub-family anywhere manufactures a pass**, which is the answer to "who decides which admissible family is sufficient?": nobody has to, because leaving a descriptor out is strictly against our own interest. What this does *not* cover is descriptors outside the catalogue entirely, which is Proposition 3's subject and Definition 1's family-relativity. **Reproduction**: `python3 run_r98_family_robustness.py` (< 1 s, no training, no GPU).

AU. Composition at Depth 8, and Three Kinds of Novelty (tag `r93_deep_composition`)

The objection this run exists to answer. The round-17 review named the composition experiment as *"the biggest weakness of the empirical story — four primitives, 200 classes, depths 1–4"* and asked, if there were compute for only one thing, for *"depths 1–8, multiple held-out composition patterns, with systematic leave-one-primitive-out and leave-one-composition-family-out"*. This run is that request, and it supersedes the depth-1–4 ladder of Appendix AH, whose construction and closure guard it inherits unchanged. It adds the condition the earlier design could not express: an unseen **arrangement** of primitives every one of which was trained, individually *and* in composition, which is what separates *learned composition* from *learned these templates*.

Why depth 8 is bought with repeats, and why that is the corpus's limit rather than our budget. `poly8` admits five rewrite primitives and only four of them can be repeated. `_rw_reassoc` and both removal schemas fire on **0** of 1102 anchors (Appendix AS); `_rw_distribute` fires on 71 of 300 measured anchors but has two or more *distinct* $\Delta{+}4$ sites on **0** of them, so it cannot be used twice in one chain. Depth 8 is therefore reached by allowing each of the four locals at most *twice at distinct sites*: a change to the chain builder, not to the corpus and not to `equivalence.py`, which is untouched here as in `r81`/`r91`. The non-local question is measured separately and reported in Appendix AS, not folded in here.

The construction, and what makes the three arms comparable. One class set of 200 runs through every depth and every arm, so every contrast is paired per class: 305 anchors were scanned and 105 dropped (58 admit no shape-matched twin, 24 lack a distinct site for some primitive, 18 cannot fill the depth ladder, 5 fail the composite constraint). Each class draws a slot sequence over the multiset $\{\texttt{commute}^2, \texttt{double_negate}^2, \texttt{add_identity}^2, \texttt{mul_identity}^2\}$, giving **2520** distinct orders, of which 188 are realised across the 200 classes and no chain is shared by more than 3. **Size matching is order-invariant by construction and recomputed per class in the verifier**: the slot deltas are $\{0, 0, 4, 4, 4, 4, 8, 8\}$, so the depth-$d$ delta is the sum over the drawn slots, and at $d{=}8$ it is $+32$ tokens for *every* class and every order, no order can be told from length. 12 arms are checked for leakage and **no training form coincides with a held-out form in any of them**.

Table 33: **The depth matrix, and the twin at every depth** (`poly8`, 200 classes, 5 seeds, chance 0.005; tag `r93`). Each row is one training library (*singles only* means the encoder never saw a composition) scored on held-out compositions at every depth. Twin rows ask which of a held-out composition and a non-equivalent partner *matched on token multiset, operator multiset and tree-local shape* is placed closer to the class centroid (chance 0.500), so they cannot be read as a length contrast. Rows below the rule are non-learned and run on the same forms.

	$d{=}1$	$d{=}2$	$d{=}3$	$d{=}4$	$d{=}5$	$d{=}6$	$d{=}7$	$d{=}8$
singles only (never composed)	.997	.983	.953	.905	.888	.841	.783	**.736**
depth-2 composites in library	.999	.993	.972	.956	.932	.914	.889	.859
depth-4 composites in library	.996	.989	.984	.979	.959	.954	.938	.919
depth-8 composites in library	.995	.989	.980	.979	.962	.961	.953	**.936**
twin, singles only	.998	.984	.980	.959	.924	.921	.894	**.876**
twin, depth-8 in library	.995	.987	.986	.975	.970	.955	.950	.949
untrained encoder ($\mathcal{X}$)	.381	.301	.195	.165	.159	.135	.119	.103
strongest non-learned (bag)	.105	.115	.100	.100	.100	.100	.100	.100
tree-local bag, the strongest *structural* bag	.055	.065	.040	.035	.060	.045	.030	.035
sup $\mathcal{F}_3$ (WL, ϕ_d, $\phi_{d\geq 4}$)	.040	.035	.020	.010	.005	.005	.005	.005
ϕ_∞, *complete* ($\mathcal{X}$)	.015	.015	.020	.005	.005	.005	.005	.005

Three kinds of novelty, priced against one covered arm. All three costs are paired differences from the depth-8-in-library row at $d{=}8$ (0.936 $[0.909, 0.961]$), on the classes each arm actually withholds. **(i) Never having composed** costs $+0.200$ $[0.163, 0.237]$: the encoder trained only on single applications still identifies depth-8 compositions at 0.736 $[0.693, 0.777]$, $147\times$ chance and $7.4\times$ the strongest zero-parameter method. **(ii) An unseen arrangement**, an ordered adjacent pair of primitives occurring in *no* training chain, with every primitive still trained individually and in composition, costs $+0.015$, $+0.011$ and $+0.010$ for the three withheld bigrams, **every interval covering zero** ($[-0.023, 0.054]$, $[-0.023, 0.048]$, $[-0.018, 0.037]$; 104, 104 and 103 member classes). **(iii) A primitive held out of the library** costs $+0.664$ on average, and never less than $+0.429$. Two things follow, and the second is against us: the gap the paper leans on is **arrangement versus inventory**, $54\times$; the gap between *depth* and inventory is only $\mathbf{3.3}\times$, so the earlier abstract's "order of magnitude", taken from `r89`'s depth-4 numbers, **does not survive at depth 8 and has been corrected to** $3.3\times$.

The branch the run committed to before the prose existed. Four readings were written into the module docstring before training started: (1) depth costs far less than primitive novelty and arrangement is cheap too; (2) the arrangement cost is *comparable* to the primitive cost, in which case the constructive claim narrows to *composition-in-seen-arrangements* and the paper is renamed;

Table 34: **Leave-one-primitive-out and leave-one-composition-family-out at** $d{=}8$ (tag `r93`). Every arm is scored against the same covered arm on the same classes, so each cost is paired. LOPO removes a primitive from the library entirely; FAMILY removes only an ordered adjacent *pair* from the training chains, leaving both its primitives trained singly and in composition.

arm	withheld	$d{=}8$	paired cost
covered	—	.936	—
LOPO	`double_negate`	.091	$+0.845\ [0.808, 0.881]$
LOPO	`mul_identity`	.231	$+0.705\ [0.652, 0.757]$
LOPO	`add_identity`	.260	$+0.676\ [0.621, 0.730]$
LOPO	`commute`	.507	$+0.429\ [0.370, 0.489]$
FAMILY	`add_identity>commute`	.921	$+0.015\ [-0.023, 0.054]$
FAMILY	`commute>mul_identity`	.939	$+0.011\ [-0.023, 0.048]$
FAMILY	`mul_identity>commute`	.924	$+0.010\ [-0.018, 0.037]$

(3) the depth cost is no longer small, withdrawing §4.3's depth claim above $d{=}4$; (4) the twin collapses at depth ≥ 6 while identification does not, making the deep result partly a size contrast and reported as one. **The run selected branch (1)**, on the run's own pre-registered test (primitive $> 3\times$ each of the others) and with the twin holding at 0.876 at $d{=}8$, which is what rules out (4). Branch (2) was the one we expected to have to write.

Membership in $\mathcal{F}_3$ is tested at every depth, and two candidates fail it, which is the point. The twin is the operational invariance test of §3.2, so it is run on all 14 non-learned members at all 8 depths. Twelve are pinned at **exactly** 0.500 on every instance and are admissible. The two order-*sensitive* ones are not: the token n-gram reaches 0.610 at $d{=}8$ and the ordered canonicalisation ϕ_∞ 0.560, so both are reported as extensions $\mathcal{X}$ outside the criterion. **Without them the** 0.500 **would be uninterpretable**: it is the pair of numbers that shows the twin is discriminable at all, rather than a task on which nothing can score.

What the guards certify, and what they do not. Three disclosures, none of which the prose leans on. *(i) Depth 8 is a property of the construction, not of the closure search.* The builder applies eight rewrites at eight verified-distinct sites, and the search-based guard finds 0 held-out forms inside the library's $(d{-}1)$-closure at every depth, but the search runs to radius 3 with a per-level frontier cap of 150, and **all** 40 **sampled searches hit that cap** (mean closure 3130 forms). Past radius 3 the certificate comes from the search-free token bound instead, which licenses $\mathbf{4}$ applications at $d{=}8$ rather than 8 ($+32$ tokens, at most $+8$ per application). **So the independently certified floor at $d{>}4$ is four rewrites, and that is what we claim**; the verifier asserts the certificate sequence $1, 2, 3, 4, 4, 4, 4, 4$ so this cannot be quietly upgraded. *(ii) The strongest non-learned row is a tie floor, not a structural cue.* `bag_variable_only` sits at exactly 0.100 at every depth from 3 on, a constant no other member shows, which is the behaviour of a feature that partitions the 200 classes into a fixed set of tied groups under cosine `argmax`. We nevertheless quote it as the skyline because it is the largest, i.e. against ourselves; the strongest *structural* bag is the tree-local bag at 0.035. *(iii) Three bigrams, not twelve.* The arrangement result rests on the three most frequent adjacent pairs, chosen by frequency so that each has >100 member classes; it is not a sweep of all ordered pairs, and all three involve `commute`.

Two results that cut against the paper's earlier reading, reported because they do. (i) `commute` is much cheaper to hold out than the other three (0.507 against 0.091–0.260). Its token delta is 0, so a library without it still contains forms of every size the ladder produces, and an encoder can partly infer sibling exchange from the other primitives' insertions. **Coverage cost is therefore primitive-specific here too**, as Appendix AS found for the local/non-local swap. **(ii) The depth cost is real and its interval excludes zero**, so §4.3 says training buys invariance to *arrangement* and only *much of it* to depth, not invariance to depth outright. The curve degrades monotonically from 0.997 to 0.736 across the eight depths on the never-composed arm; what is flat is the *covered* arm ($0.995 \to 0.936$).

And one earlier non-replication, now withdrawn, along with the explanation we offered for it. Earlier revisions noted that Appendix AS flagged an alternate-order arm costing $+0.250$ at $d{=}4$ against nothing here, and offered *training-order diversity* as the consistent reading: `r91`'s library holds one order, while each arm here trains on 400 chains of which 341 are distinct. **Both halves of that were wrong, and we say so with the numbers.** The $+0.250$ was an aggregation artifact in `r91` and is corrected there; within a fixed training library that run's order cost is at most 0.035 and negative as often as not, so **there was never a discrepancy for diversity to explain**. And when we ran the experiment this appendix said it had not run, diversity turned out not to be a mechanism either: holding class set, site filter, depth, anchors and scored forms at `r91`'s values and varying *only* how many distinct orders the training composites span, order diversity removes -0.0075 $[-0.033, 0.018]$ of the cost on ladder L and $+0.0075$ $[-0.014, 0.026]$ on ladder N (tag `r94`; Appendix AV). **The honest summary is that the arrangement result needed no scope condition and we had proposed one to explain a number that was not real**, the reading was consistent with the evidence, which is exactly why offering it as a reading rather than a finding was the right discipline and why running it was still necessary.

Reproduction. `python3 run_r93_deep_composition.py --K 200 --max_depth 8 --num_seeds 5` (183.5 min on one Apple M-series GPU; `REPRODUCE.md` R29). Eleven trainings per seed (four library depths, four LOPO arms and three FAMILY arms) with every twin and every non-learned row scored on the already-trained encoders, so the twin costs no additional training.

AV. Does Training-Order Diversity Explain the Alternate-Order Gap? (tag `r94_order_diversity`)

An experiment the paper named against itself. Until this revision Appendix AU closed with a discrepancy it could not resolve, `r91` reporting a $+0.250$ alternate-order cost where `r81` and `r93` reported none, offered training-order diversity as "the consistent reading rather than as a measured mechanism", and stated plainly that *"we have not run the experiment that would separate diversity from the other differences (class set, site filter, depth), so this is not a finding."* **This is that experiment.** It returned two results: the mechanism is not diversity, and the discrepancy it was invented to explain does not exist.

Three conditions, not two, and the amendment is dated. The pre-registered design compared a single-order library against an order-diverse one. A smoke run showed that this cannot work: `r91`'s published alternate-order numbers come from its *singles-only* encoder, so a control trained on composites-in-one-order is not `r91`'s condition (it scored forward 0.977 / alternate 0.947, nothing like 0.922 / 0.671), and two conditions cannot separate *trained on composites at all* from *trained on composites spanning several orders*, which is exactly the confound the appendix admitted to. **The design was amended to three conditions before any full-scale number was read**, and the amendment is recorded in the pre-registration above the results:

singles the library of single applications only, no composites. *This is `r91`'s published condition.*
control $+n_{\text{comp}}$ composites, **all** following the forward permutation. One training order.
diverse $+n_{\text{comp}}$ composites following **distinct** permutations of the same multiset.

so SINGLES→CONTROL isolates *composition* and CONTROL→DIVERSE isolates *order diversity with composition held fixed*.

What is held fixed, asserted rather than stated. Class set (`r91`'s 133 $\Delta{+}4$-filtered classes), site filter, depth ($d{=}4$), anchors, skeletons, k_{para}, n_{comp}, seeds, architecture and **the scored held-out forms**, the alternate-order form is drawn *before* the composites and is identical across all three conditions, so every arm is scored on the same target. Both composite arms train on the same *number* of composites at the same depth: the run asserts the token delta is $+16$ for every composite in both, since a permutation of a multiset cannot move it. The manipulation itself is asserted on *orders* and not on serials: the control spans exactly 1 distinct order per class and the diverse pool 2, and the held-out form appears in 0 training composites of either arm. Only order-span differs.

Table 35: **Order diversity removes nothing, because there is nothing to remove** (`poly8`, `r91`'s K=133 classes, d=4, 5 seeds, chance 0.0075; tag `r94`). Positive cost means the alternate order is *harder*. **Every interval in the cost column contains zero**, in all three conditions and both ladders. **A null needs its sensitivity shown**: changing the training library moves the forward column by $+0.048$ (L) and $+0.078$ (N), which is more than any cost here but only $1.7\times$ on L, so the load-bearing anchor is that the *same* paired estimator on these *same* 133 classes resolves coverage effects of $+0.204$ and $+0.468$ with intervals excluding zero (Table 32).

	ladder L (all local)			ladder N (one non-local)		
training library	fwd	alt	cost (paired)	fwd	alt	cost (paired)
SINGLES (`r91`'s condition)	.929	.901	$+.029$ $[-.017, .074]$	.911	.923	$-.012$ $[-.059, .035]$
CONTROL (composites, 1 order)	.977	.959	$+.018$ $[-.009, .047]$	.990	.993	$-.003$ $[-.023, .012]$
DIVERSE (composites, 2 orders)	.970	.944	$+.026$ $[-.014, .065]$	.984	.994	$-.011$ $[-.029, .005]$
cost removed by *composition*			$+.0105$ $[-.023, .044]$			$-.0090$ $[-.050, .030]$
cost removed by *diversity*			$-.0075$ $[-.033, .018]$			$+.0075$ $[-.014, .026]$

The result, and none of the pre-registered branches applies. All four branches (diversity explains the cost, diversity does not and Figure 3's arrangement bar must be re-scoped, a partial, or composition rather than diversity being the mechanism) **presuppose that a cost exists to be explained**. None does: the largest alternate-order cost anywhere in Table 35 is $+0.029$ with an interval covering zero, and the diversity contrast is -0.0075 on one ladder and $+0.0075$ on the other, the two signs disagreeing. **We therefore report the outcome the pre-registration did not enumerate**, which is the honest description of what happened, and note that branch (2), withdrawing the arrangement bar, was the live risk this run was built to expose and is the one the numbers rule out.

The validity check became the finding, and we deviated from a pre-registered rule to let it. The pre-registration made SINGLES a cross-run check with a discard rule: it "should land near `r91`'s $+0.250$", and "if it does not, the port is wrong and the run is discarded." It landed at $+0.029$. **We did not discard the run, and the reason has to be stated rather than assumed.** The rule presumed the published number was correct, and the port in fact reproduced `r91`'s published *cells* while failing to reproduce its published *delta*: forward 0.929 against 0.922 and alternate 0.901 against 0.893, both within seed noise, yet a delta of $+0.029$ against $+0.250$. **A discrepancy that lives in the delta and in neither number it is computed from localises the fault to the aggregation**, so we audited `r91`'s log instead, and what the audit returned is arithmetic rather than judgement: all six of its published deltas equal a full-library forward score minus an alternate-order mean taken across two *different* training libraries, exactly, and the corrected ladder-L cost is $+0.0286$ against this run's independent $+0.0286$ (Appendix AS). **We record the deviation as a deviation**; had the arithmetic not closed, the run would have been discarded as specified.

Three negative controls, each firing on the assertion under test. **C1**, collapsing the diverse pool to a single order, makes the two composite conditions *numerically identical* (.917 against .917, diversity contrast exactly 0.0000 with a zero-width interval), so they differ only through the manipulation. **C2**, injecting the held-out alternate-order form into both libraries, collapses the cost to exactly 0.0000 in all three conditions, showing the measurement does respond to what it claims to measure. **C3 exists because C2 does not test the guard**: a passing `leak == 0` proves no leak was present, never that one would be seen, so C3 injects the same form *before* the guard counts, and the guard fires with the right count (`12 training composites ARE the held-out alternate-order form`).

Reproduction. `python3 run_r94_order_diversity.py` (49.6 min on one Apple M-series GPU; `REPRODUCE.md` R30). Thirty trainings, three conditions $\times$ five seeds $\times$ two ladders, plus 316 s building the paired class set. Controls: `--nc_collapse`, `--nc_leak`, `--nc_leak_pre` (the last must abort).

AW. The Auditor Ported to Code: Type-2 Clone Classes (tag `r95_code_audit`)

Why this section exists. It documents a result the body has cited since the previous revision without a self-contained section of its own, a gap this appendix's own reading map promised not to have,

since it claims one section per run tag. **The omission is part of why the port reads as thin**, and the material below is the port's guards, which are what makes it evidence of anything.

The corpus, and what it is not. 300 Type-2 clone classes, 4 members each, built by α-renaming functions drawn from the *real* Python standard library (400 files scanned, functions kept at 40–400 AST nodes, 3 renamed variants per original, seed 95). **It is constructed**: the sources are real code and the clone relation is ours, whereas a published clone benchmark would supply both. So it evidences that the auditor *ports* to a non-symbolic modality, not the prevalence of the flaw in the clone-detection literature, which this corpus cannot speak to. Appendix AX is the published-corpus counterpart added for this reason.

The mapping. **S1** (variable identity) $\rightarrow$ identifier names; **S2** (operator/arity) $\rightarrow$ AST node type paired with its arity. Nothing else changes: the same `separability_from_bags` that produces every other row of Table 23 produces this one.

Two guards, each asserted rather than reported. **G1**: the S2 bag is renaming-invariant on all 300 classes, if α-renaming moved a node-type bag, the mapping would be wrong. **G2**: identifiers *do* vary within all 300 classes, without this the S1 column would be measuring nothing, since a class whose members share identifiers is trivially separable at S1 for the wrong reason. Both are hard `asserts`; the run aborts rather than reporting a number.

Result, and the number that bounds it. T1 96%, T2 100% (Table 23), T1 firing harder than on any symbolic corpus, including AI Feynman's 84%. As everywhere in that table the T1 column measures class *representatives*; what a shortcut could actually transfer through here is the identifiers a renaming leaves shared, **measured at** 0.308 **held-out overlap**. The two numbers answer different questions and both are reported.

A correction this revision makes to the sentence above. The previous revision let that 0.308 read as a *bound* on the S1 shortcut. **It is not one, and measured directly the shortcut is far stronger**: the held-out identifier bag scores 0.770 against a chance of 0.0033 (AZ), because α-renaming cannot touch imported, builtin or attribute names, which stay shared across a class and across the corpus. The overlap statistic *understates* the S1 exposure of this corpus, and we now say so rather than leaving the inference to a reader.

Reproduction. `python3 run_r95_code_audit.py` (seconds, CPU only; no download: it reads the local Python installation's standard library).

AX. The Auditor on a Published Benchmark Outside Symbolic Mathematics: SCAN (tag `R96_SCAN_AUDIT`)

What is ours here and what is not, stated before any number. SCAN (Lake & Baroni, 2018) is fetched as published (`tasks.txt`, 20,910 commands, SHA-256 pinned in `fetch_scan.sh`, BSD licence, **not redistributed**). **The equivalence classes are SCAN's own**: two commands are equivalent iff the benchmark's own gold action sequence for them is identical, and it is SCAN's own paraphrase, `X and Y ≡ Y after X`, that populates them. 9,228 distinct gold sequences, 9,222 of them with two or more distinct *published* surface forms. **What is ours is the level mapping and the pairing criterion** for which published pairs count as twins. **So this audits the cue structure of SCAN's semantic equivalence classes; it says nothing about SCAN's published seq2seq results**, which we do not touch.

The mapping, and why no new measurement code was written. **S1** $\rightarrow$ the content words `walk look run jump turn left right`; **S2** $\rightarrow$ the connective/modifier inventory with counts, `and after twice thrice opposite around` plus the direction-attachment rule; **S3** $\rightarrow$ ϕ_d over the SCAN parse tree, children unordered, exactly as in AC; the **extension** $\rightarrow$ the same tree with children in order. A recursive-descent parser for SCAN's published grammar emits the paper's own `Node` type, after which `phi_d_bagger`, `wl_subtree_bagger_stable` (the digest-based variant, so the WL rows do not depend on the interpreter's hash seed), `canonical_tree_bagger`, `var_multiset`, `op_multiset`,

`token_multiset`, `separability_from_bags` and `bag_classifier_accuracy` run *unchanged*. **The only new code is the adapter, not the measurement.**

Five guards, and the two that matter fire on populations we declare. **G0**: all 20,910 published commands parse and are fully consumed, and each has exactly one gold output, so the class relation is a partition. **G1**: every audited class has ≥ 2 distinct published surface forms. **G2**: held-out-form leakage $0/200$. **G3, admissibility, per instance**: across 37,758 published twin pairs (non-equivalent, matched on the word multiset and on every $\mathcal{F}_3$ member), all eleven order-blind candidates are pinned at *exactly* 0.500. **G4, the control that makes G3 capable of firing**: on the 22,584 of those pairs whose ordered form differs, the order-sensitive completion scores 1.000 and is **rejected** from the family as an extension. **G5, saturation**: the corpus's maximum parse depth is 4, and ϕ_8 and the order-blind canonical form induce the *identical* 168-block collision partition over all 20,910 commands, asserted as partition equality, not argued.

A bound the split exposes, which we report rather than filter away. The remaining 15,174 twin pairs are ones the ordered completion *cannot* separate either: they differ only in which content word sits at which position, which no bag in this paper reads. **On that population every candidate here, ϕ_∞ included, is pinned at** 0.500**.** Discarding those pairs would have made G4's 1.000 an artefact of our own pair selection, so both populations are reported with their sizes.

The pre-registered prediction, and the branch that was observed. Written before the first run: if the non-monotone ceiling is a property of admissible families rather than of polynomial algebra, the coarsest member should attain the ceiling here and the ordered completion should sit below it. **It does not.** At $K{=}200$ (chance 0.005) the measured held-out ladder is ϕ_d 0.080 $\rightarrow$ 0.115 rising and then flat, WL_h 0.040/0.070/0.095 rising in h, the family ceiling 0.115 at $\phi_{d\geq 2}$, and the ordered completion *above* the family at 0.155. **We report this as a domain boundary**, which is the branch the pre-registration names for this outcome; the corpus was not reshaped and no row was dropped.

The null's sensitivity, because a null needs one. The verdict is unchanged across the same K ladder the polynomial corpora are reported at, three seeds each: in 9 **of** 9 **cells** the inversion fails to replicate, ϕ_d is monotone in d, and the ordered completion is never below the ceiling. At $K{=}1000$ the whole $\mathcal{F}_3$ family collapses to 0.001–0.003 against a chance of 0.001, while a content bag holds 0.077–0.079.

What the audit diagnoses instead, and the divergence it sharpens. On SCAN the cue is *lexical*, not structural: at $K{=}200$ a content bag scores 0.320 and a token bag 0.435 against a family ceiling of 0.115 and chance 0.005. **And the S2 column is the sharpest divergence between the survey statistic and the measured number anywhere in this paper**: pairwise operator-bag separability is 98.8%, which reads as maximal exposure, while the measured held-out accuracy of that same bag is exactly 0.000, because SCAN's own paraphrase *changes* the connective, making the S2 bag anti-informative about the class. Table 23's caveat that its columns measure representatives rather than held-out forms is not a hedge; this is what it looks like when it bites.

Reproduction. `./fetch_scan.sh` then `python3 run_r96_scan_audit.py` ($\approx$ 3 min, CPU only, no GPU and no training). The survey row of Table 23 is `separability_from_trees(..., max_classes=600, seed=0)`, the survey's own call and its own subsample convention, so the row is computed exactly as the other 23 are.

AY. A Composition Split We Did Not Design: SCAN `add_prim_jump` (tag `r97_scan_composition`)

The objection this run exists to answer, and the previous revision's answer to it. Every design choice behind §4.3 is ours: which primitives compose, at which sites, to which depth, and which forms are held out. A reader can grant every number and still say the composition result is a property of a protocol we built. The previous revision said in print that an independently designed composition benchmark would test this and that we did not run one. **This is that run**, and it uses a corpus already in this paper rather than a new one: SCAN's own `add_prim_split` (Lake &

Baroni, 2018), whose published train/test files are SHA-256 pinned in `fetch_scan.sh` alongside the `tasks.txt` that Appendix AX audits.

What is theirs, itemised, because that is the whole point of the run. **Theirs**: the grammar, the transformations, the equivalence relation, and the train/test split. **Ours**: the admissible family and the audit question. The composition to be generalized, `jump` appearing only as a bare primitive in training and only inside composed commands at test, is Lake and Baroni's design decision, not ours, and we neither reshaped the corpus nor re-drew the split.

The split, measured from the published files rather than described. 13,204 train commands over 5,627 multi-form classes; 7,706 test commands over 3,595 multi-form classes; 20,910 commands in total, so the two files partition the corpus exactly as `tasks.txt` does. **Surface overlap 0 and class overlap 0**, both asserted rather than assumed. `jump` occurs 1,467 times in training and in every one of them as the bare primitive; it occurs in all 7,706 test commands, and all 3,595 test classes are `X and Y` $\equiv$ `Y after X` pairs containing it. Every command parses and 0 fail; maximum parse depth is 4, as in Appendix AX.

Chance from the observed distribution, never $1/K$. Test class sizes are uneven (3,371 of size 2, then 9, 183, 7, 15, 4, 4 and 2 at sizes 3 through 10), so the null is the exact random-label expectation 0.000171 and not $1/3595 = 0.000278$; the latter is stored in the log under `superseded_1_over_classes` so the two can never be confused. A 2000-draw label permutation reproduces it at 0.000166 $[0.000000, 0.000649]$.

Two guards, and the control that makes the second capable of firing. **Leakage**: 0 held-out forms and 0 held-out classes appear in training. **Admissibility, per instance, on their test split's own twins**: 14,900 non-equivalent pairs matched on the word multiset, split as $\mathbf{14{,}900 = 8{,}852 + 6{,}048}$ into pairs whose ordered form differs and pairs differing only in which content word sits where. The sum is asserted, because a twin filter chosen after the fact can make a guard's score an artefact of one's own pair selection. All eleven order-blind candidates are pinned at *exactly* 0.500 on both populations. **The positive control**: the order-sensitive completion scores 1.000 on the 8,852 order-differing pairs and 0.797 over all 14,900, and is **rejected** from the family, a membership test no candidate can fail would prove nothing.

The family, and the result. The ceiling is $\sup \mathcal{F}_3 = \mathbf{0.0005}$ $[0.0001, 0.0010]$, attained by ϕ_{d1} and matched to four decimals by every other member; the strongest non-learned representation of *any* kind here, admissible or not, is a token bag at 0.0167 $[0.0112, 0.0226]$, and the S2 operator bag is at 0.0005 because SCAN's paraphrase changes the connective (Appendix AX). A Tree-LSTM trained on the published train split, three seeds, scores 0.9799 $[0.9759, 0.9835]$, 0.9864 $[0.9830, 0.9894]$ and 0.9940 $[0.9918, 0.9960]$: mean $\mathbf{0.9868}$, worst-seed lower bound 0.9759, and **every seed exceeds both the family ceiling and the token bag with disjoint intervals**.

The prediction was registered before the run and it held, and the previous SCAN run's did not. Registered: *a trained encoder on the published train split exceeds* $\sup \mathcal{F}_3$ *on the published test split, with disjoint* 95% *class-bootstrap intervals.* Confirmed. **We record that this is the second pre-registered SCAN prediction and the first to hold**: Appendix AX's prediction failed and was reported as a domain boundary. Both branches were named in advance in both runs.

Why we quote no ratio, and what the ratio is. $0.9868/0.0005 = 1973.6$. **We do not put that number in the body, and the reason is the paper's own argument**: SCAN's held-out design puts the admissible bar on the floor, so the ratio measures the corpus's construction and not the encoder. §4.2 declines the flattering $6.1\times$ against a variable-blind bag for the same reason, and it would be incoherent to accept a $1973.6\times$ here. The honest statement is interval disjointness plus this disclosure.

One measurement caveat, disclosed rather than filtered. Three of the twelve non-learned representations (WL_{h1}, WL_{h2}, WL_{h3}) have an unclean tie valley at the 10^{-6} quantisation step (1, 2 and 5 ambiguous gaps respectively), so their k-NN scores could in principle shift at the fourth decimal. **The ceiling does not rest on any of them**: it is attained by ϕ_{d1}, whose valley is

clean, and all three WL rows score the same 0.0005 regardless. The count is stored in the log as `n_representations_unsafe_tie_valley`.

What this does and does not license. It removes the circularity: the composition finding survives on a split whose transformations, grammar and held-out set are someone else's. It does **not** widen the claim. What clears the bar here is composition of *one known* transformation (`and`/`after`) onto *one* unseen primitive (`jump`), narrower than §4.3's depth-8 chains, and still not compositional reasoning. And the bar it clears is a low one, which is why the previous paragraph exists.

Reproduction. `./fetch_scan.sh` then `python3 run_r97_scan_composition.py` (≈ 33 min on MPS; 10 epochs, 3 seeds, k=1, embedding dimension 128, margin 0.2). Every measurement routine is reused unchanged from `r70` and `r96`, the identification protocol, k-NN scorer, class bootstrap, permutation null, twin builder and the whole `structural_baselines` registry. **The only new code is the split loader.**

AZ. Finishing the Code Ladder to S3, and a Task the Audit Rules Settled (tag `r99_code_structural`)

Why this run exists. Appendix AW ported the auditor to code at S1 and S2 only. A two-level port is not an instantiation of a four-level criterion, so the generality claim rested on SCAN alone. **This run finishes the ladder on the *same* corpus** (same functions, same clone classes, same seed, nothing retrained and nothing downloaded) so that the second modality carries a full audit rather than half of one.

What is reused and what is new, stated before any number. **Reused**: `collect_functions` and `build_clone_classes` from `run_r95_code_audit` (the class count is asserted equal to `r95`'s 300), and `phi_d_bagger`, `wl_subtree_bagger_stable`, `canonical_tree_bagger` and `bag_classifier_accuracy` unchanged from `structural_baselines` and `benchmark_audit`. **New**: an adapter from Python `ast` to the paper's `Node` type, and nothing else. **Two properties of that adapter are ours and not the data's, and both are constructions**: identifiers and literals are anonymised (the symbolic pipeline's convention, and the conservative choice here, since it is what makes S3 variable-blind); and because `Node` is binary, an n-ary AST node becomes its first child paired with a **right-fold of the rest under a `seq` spine**. **So this corpus is constructed in two independent ways**, the clone relation (Appendix AW) and now the tree shape, and neither supports a prevalence claim about the clone-detection literature. Adapted trees run 34–361 nodes, mean 112.8, on Python 3.14.6.

Three guards, and the positive control that makes the third capable of firing. **G1** and **G2** are `r95`'s, re-asserted here: the S2 node-type/arity bag is renaming-invariant on all 300 classes, and identifiers *do* vary within all 300. **G3** is new and is the mechanism behind the predicted result: the variable-anonymised trees of co-clones are *identical* in all $\mathbf{300/300}$ classes. **The positive control**: run with `--pc_no_anonymise`, which keeps identifiers, and G3 drops to $\mathbf{0/300}$ and the run aborts. A guard that cannot be made to fail proves nothing, and `leak == 0` style passes have twice been silent here.

The attainable maximum, computed rather than assumed. Nearest-centroid classification cannot separate two classes whose held-out representatives are identical, so the maximum any representation can reach is fixed by the corpus. Of the 300 anonymised representatives, **297** are distinct; the 3 duplicates fall in three groups, `tty.py::setraw`/`setcbreak`, `turtledemo/penrose.py::kite`/`dart`, `heapq.py::heappop`/`heappop_max`. **These are real standard-library functions that are themselves Type-2 clones of each other, which the construction split into separate classes**: an incidental labelling error in a constructed benchmark, found by the audit and reported rather than filtered. The attainable maximum is therefore exactly $(300-3)/300 = \mathbf{0.990}$, and the assertion in `verify_claims.py` is that equality, not a threshold.

The pre-registered prediction, and the ladder. Registered before the full run: $\sup \mathcal{F}_3$ *equals the attainable maximum and is attained at every* d *and every* h, because α-renaming leaves the tree

untouched (G3). **It held.** Against a chance of 0.003333: the S1 identifier bag 0.770, the S2 node-type/arity bag 0.9867, and *all seven* $\mathcal{F}_3$ members ($\phi_{d\in\{1,2,4,8\}}$ and $\mathrm{WL}_{h\in\{1,2,3\}}$) at 0.990, i.e. all seven at the maximum. **The order-sensitive extension ϕ_∞ gets no further either**, also 0.990, so there is no headroom above S3 for anything to occupy.

The verdict, which is the point of the run. Type-2 clone detection on this corpus is an S1–S3 task: the declared admissible family already reaches everything the corpus allows, so *no* score on it, ours or anyone's, can be evidence about structure above S3. This is a fourth distinct outcome of the audit, alongside a pass (§4.2), an entitlement correctly identified (Appendix AR) and a domain boundary (Appendix AX), and it is what it looks like when the criterion *declines to certify a task* rather than a model. **We report it as a verdict of the audit and not as a result in our favour.**

Two things deliberately not run. Tree-edit distance is quadratic in node count and these trees reach hundreds of nodes over 300 classes; the ordered completion ϕ_∞ already bounds the order-sensitive extension, and it is at the maximum. **Nothing is trained in this run, by design**: the finding is a statement about the corpus's cue structure, and a trained encoder could not exceed 0.990 either.

Reproduction. `python3 run_r99_code_structural.py` ($\approx$ 2.0 min, CPU only; no download: it reads the local Python installation's standard library, as `r95` does). Positive control: `--pc_no_anonymise` (must abort).

BA. The Admissible Ceiling Over *Readouts*, Not Just Over Feature Maps (tag `r100_learned_invariant`)

Why this run exists: the instrument was under-measuring the family we had declared. Definition 1 declares the audit statistic to be $\sup\{M(g) : g \in \mathcal{F}_t\}$ over representations *determined by the cues at levels $\leq t$ alone*. **A trained readout over an order-blind feature map is determined by those cues alone, so it *is* a member.** Every ceiling this paper had published, however, was measured with *one* fixed, unfitted nearest-centroid readout (`benchmark_audit.py`: the held-out form is scored against class centroids in bag space). **So every printed** $\sup\mathcal{F}_3$ **but the twin's was a lower bound on our own statistic**, and the gap was unmeasured. That is a defect in the instrument rather than a positioning problem, and this appendix reports its repair in three parts: admissibility *proved* for a trained comparator, the ceiling re-measured with the readout fitted, and (the load-bearing part) a bound on the supremum over **every** readout at once, computed with no training at all.

Part A: a *trained* comparator admitted by proof, and the negative control that gives the proof content. Protocol: the shape-matched sibling swap on the *published* $K{=}500$ split (348 classes, chance 0.500), unchanged. Input maps: the seven of Definition 1 plus the tree-local bag, each feeding an `InvariantHead` (Linear $\to$ ReLU $\to$ Linear, ℓ_2-normalised) trained with the paper's own semi-hard triplet loss. **Because every input map is order-blind, a held-out form and its twin have *identical* input vectors** (asserted, not assumed) **so the comparator's twin score is** 0.500 **for *any* parameters.** All eight sit at **exactly $\mathbf{0.500}$**, pinned *per instance* rather than on average, under **both** an untrained and a trained head: this is the family's first member whose admissibility is *proved* rather than tested. **The negative control moves in the other direction.** The order-sensitive n-gram bag has non-identical twin inputs, is not pinned, and **training *raises* its twin score,** $0.614943 \to 0.764368$, training makes an *inadmissible* comparator better at the very property that disqualifies it, and changes an admissible one's twin score by exactly 0. **That contrast is the answer to "is this just *use an invariant control*"**: admissibility is a property of the comparator's *input pipeline*, not of its strength or its parameter count, and here the two dimensions are measured apart rather than argued apart.

Part B: the ceiling rose at every fitted cell, as pre-registered. Registered before the run: *the fitted readout raises the ceiling everywhere, because a fitted metric on the same features cannot do worse than a fixed dot product*. **It held at all seven fitted cells**: on `poly8` $+.042/+.055/+.041/+.008$ at $K{=}50/100/200/500$, on `boolean8` $+.106/+.091/+.107$ at $K{=}50/100/190$, largest rise $+\mathbf{.1071}$, and the fit helped all eight maps in every cell. **Regression control**: the *published* ceiling is reproduced by the *unfitted* readout on the same partition, 20 rows against

r71_tier3_baselines_v3 and 15 against r87_boolean_tier3, 0 mismatches, so a rise is attributable to the fitted readout and not to a changed pipeline. The 12 and 9 WL rows are carried as *advisory* rather than dropped, because Python's salted hash() makes those feature spaces process-dependent (Appendix AI, the same exception r92 records); their maximum absolute deviation is 0.0.

Part C: the supremum over *every* readout, with nothing trained. A readout cannot separate what its input has already merged. So the **collision partition** of the held-out forms induced by an admissible map, forms with identical feature vectors in one block, bounds *any* readout over that map from above: the best achievable is to label each block by its most frequent class. **The structural fact that makes this a bound over the catalogue rather than over one map**: of the 13 admissible maps, **12 induce exactly the same partition in every cell and the 13th (the Laplacian-spectrum bag) is strictly *coarser*** (named, not averaged away) so **no admissible map is finer**, and the bound at the common partition bounds them all. **Positive control**: the order-*sensitive* canonicalised-ordered completion is strictly finer with a strictly higher bound in all eight cells (85 blocks against 34 at poly8 $K{=}200$, bound .635 against .542), **the only maps that refine the partition are the ones the twin test rejects**, which is the closure property showing up as a measurement. The n-gram control is *incomparable* in six cells and strictly finer in two; we report the split rather than the majority relation.

Corpus	K	$\sup \mathcal{F}_3$ as published one fixed readout	fitted readout Part B	**bound over *every* readout** Part C, untrained	encoder class lower bd. published	readout-proof?
poly8	50	.7355	.7774	.7914	.6126	**no**
poly8	100	.6408	.7039	.7322	.7604	yes
poly8	200	.4883	.5338	.5417	.8007	yes
poly8	500	.2763	.2843	.2962	.8572	yes
poly8	1000	—	—	.3385	.8200	yes
boolean8	50	.7320	.8380	.8733	.8706	**no**
boolean8	100	.5250	.6155	.6523	.9574	yes
boolean8	190	.5113	.6184	.6400	.9251	yes

What the bracket settles, and what it does not. Every bracket is consistent (published $\leq$ fitted $\leq$ bound, asserted cell by cell), and the bound sits strictly below the metric's attainable maximum in all eight cells, so it is not vacuous. **Six of the eight cells are readout-proof**, the trained encoder's class-level lower bound exceeds the supremum over *every* admissible readout, so no member of $\mathcal{F}_3$ at any capacity can account for its score. **The two that are not are exactly $K{=}50$ on both corpora**, and they are not equivalent: poly8 $K{=}50$ sits inside the range the paper already declines (§4.2 scopes the claim to $K{\geq}200$), while **boolean8 $K{=}50$ does not, and is narrowed here**, .8706 against a bound of .8733, **one of our own pass verdicts left unproved by** .0027. We state it as the margin it is rather than as a rounding. **From poly8 $K{=}100$ upward the bound excludes every admissible readout**, one rung *below* the $K{\geq}200$ the body scopes its claim to, so **the scoping the paper already declared is the stricter of the two, and the wider statistic corroborates it rather than forcing it to be loosened**. **This is the consequence the obvious principle does not have**: "use an invariant control" cannot bound a comparator nobody ran, and a family closed under post-composition can.

Appendix AO's untrained-encoder anomaly splits in two, and one half of it was ours. The untrained Tree-LSTM's boolean8 scores are r86's, so this is a cross-log comparison. At $K{=}50$ the *fitted* ceiling .838 **overtakes** the untrained encoder's .813: that half was a *readout* artifact, and attributing it to the family was wrong, and the readout-free bound .8733 does not exclude a member matching it either, the same cell and the same cause as the unproved pass above. At $K{=}100$ and $K{=}190$ the opposite holds *provably*: .769 and .698 exceed the readout-free bounds .652 and .640, so **no member of the declared family, under any readout, can reach the untrained encoder there**. That half is a property of $\mathcal{F}_3$ on this corpus, established by proof rather than by exhausting comparators.

Disclosed rather than filtered, and why the paper leans on Part C. The k-NN tie valley is *unsafe* for every seed of every attaining member on boolean8 (bag_wl_h3 at $K{=}50$ and $K{=}190$, $\phi_{d=1}$ at $K{=}100$), so those fitted values depend on a tie-breaking convention. **And the training loss did not fall at every poly8 map at $K{=}200$ and $K{=}500$** (all three seeds returned the identical .5338 at $K{=}200$) while it fell at every boolean8 map at every cell. **Both are why**

the body leans on Part C's bound rather than on Part B's fit: the bound needs no ranking, no tie rule and no training, and it is the stronger statement in any case. **One asymmetry is disclosed rather than hidden**: `poly8` $K{=}1000$ has a bound but no fitted row, because Part B stopped at $K{=}500$ and Part C did not; the verifier asserts that cell list at its actual value instead of at the empty list.

Reproduction. `python3 run_r100_learned_invariant.py` ($\approx$ 3 h 29 min $=$ 12513 s, one Apple M-series GPU). **No encoder is trained**: one `InvariantHead` per (corpus, K, feature map, seed), 400 steps, learning rate 10^{-3}, margin 0.2, seeds 97/98/99, `shuffle_seed` 70, so the partition is `r70`/`r71`'s published one, field for field, and Part C trains nothing at all.

BB. An *Adversarial* Search for an Order-Blind Descriptor That Wins (tag `r101_adversarial_family`)

Why this run exists, and why it is not another widening. The standing objection to a *declared* family is that the audit rules out only the explanations we chose in advance. Appendix AT answers it by hand (five descriptors we did not choose), and a hand-picked widening is still a list somebody wrote. **This run answers it by *search***: an adversary whose objective is to beat the encoder, allowed to *compose* descriptors rather than pick from a list, given **oracle access to the very split we report**, and required only to pass the same per-instance twin test every declared member passes. **Three advantages are handed to it deliberately**; it selects on the reported partition, it composes, and it starts from the strongest control the paper publishes, because an adversary that loses without them establishes nothing.

Why the search does not run at the twin. At the shape-matched twin the ceiling is not empirical: Proposition 4 shows that *every* arrangement-invariant representation ties on every trial and therefore scores exactly 0.500, so a supremum over $\mathcal{A}_P$ is known there by proof and a search could only rediscover the value, and, being a maximisation, could only ever *lower*-bound what is already an equality. **So the search runs at the rung where the ceiling really is an empirical question**: `poly8` $K{=}500$ UnseenEqClass, the cell §4.2 prints.

Three reference lines, separated *before* the search, because the paper prints two different numbers. **L1**, the supremum over the seven maps Definition 1 enumerates: $\mathbf{0.2763}$, and all seven attain it. **L2**, the supremum over weighted *unions* of those seven: each union is determined by the cues at levels ≤ 3 alone, so nothing about it is inadmissible, yet Definition 1 does not enumerate it: **this, not L1, is the line the printed** 0.276 **is exposed to**. **L3**, the supremum over every order-blind descriptor the twin test admits, whose published value is the full-token bag's $\mathbf{0.5174}$, the line §4.2's ratio is read against, and the one the objection means. **L3 is the hard line, and it is the one that moved.**

What the adversary was allowed, stated as a search space rather than as a result. 28 order-blind primitives: the seven declared, ϕ_d at $d \in \{3, 5, 6, 7\}$, WL_h at $h \in \{4, 5, 6\}$, the full-token, variable-blind, variable-only and tree-local bags, the canonical *unordered* tree bag, graphlet bags of order 3, 4, 3–4 and 3–5, a root-to-leaf path bag, and Laplacian-spectrum bags at 1–4 eigenvalues. Candidates are weighted unions of at most 6 primitives, component weights reached multiplicatively over $\{\frac{1}{16}, \ldots, 32\}$; greedy forward selection then weight climbing, from the strongest singleton and from three random size-3 starts, improvement threshold 10^{-4}. **706 distinct composites were scored in** 46 **ladder calls**, each scored by the shipped `r71` ladder with the bagger registry swapped, so no scoring code is new. **Nothing is trained anywhere in this run.**

The admissibility gate, and the two controls that make it capable of firing. Appendix AT's per-instance test, verbatim: the published $K{=}500$ swap split, 348 classes, chance 0.500, and **a member must be pinned at exactly** 0.500 **on *every* instance, not on average**. **Positive control**: `bag_tree_local` pins at 0.500. **Negative control**: the order-sensitive `bag_token_ngram` reads 0.681034 and is not pinned, so it is reported as an extension $\mathcal{X}$ and excluded, the same treatment ϕ_∞ and tree-edit distance get. Both controls behaved. **All** 28 **primitives were admitted and none rejected**, and the two discovered winners were *re-gated after* the search and pinned at exactly 0.500 as well: what the search found is an admissible control, not an extension that cheated.

Track 1: the pre-registered prediction held, and it strengthens a printed number. Registered before the run: *no union or reweighting of the declared seven beats the best single member*, because `poly8`'s trees are depth ≤ 4, $\phi_{d\geq 4}$ is already the complete order-blind fingerprint, and the seven's collision partitions coincide. **It held exactly**: all 67 candidates (every union, every reweighting, four independent local optima) read $\mathbf{0.2763}$, so $\mathbf{L2} = \mathbf{L1}$ to four decimals with nothing anywhere above. **So the printed** 0.276 **is the supremum not merely over Definition 1's enumeration but over every weighted union of it**, and Appendix BA's readout-closed bound 0.2962 is undisturbed.

Track 2: the pre-registered prediction did *not* hold, and that is this appendix's result. Registered: *the search would not materially exceed the published* 0.5174. **It did.** The best composite reads $\mathbf{0.6758}$, class bootstrap $[0.6096, 0.7351]$, **a rise of** $+0.1584$ **over the strongest control the paper had published**, and it is

$$\mathrm{bag}\big[\phi_{d=6} \;+\; 8 \cdot \mathrm{tok}_{\mathrm{full}} \;+\; \mathrm{tok}_{\mathrm{varblind}} \;+\; 16 \cdot \mathrm{tok}_{\mathrm{varonly}} \;+\; \mathrm{WL}_1 \;+\; \mathrm{WL}_4\big],$$

six components, pinned at exactly 0.500 on the twin when re-gated. **It is not one lucky basin**: two unrelated random restarts reached 0.6577 and 0.6523, and the greedy run from the strongest singleton reached 0.6635. **The mechanism is not mysterious and is worth naming**: every gain comes from up-weighting descriptors that read the *variable identity* ϕ_d anonymises away, which is exactly why the full-token bag outscored $\sup \mathcal{F}_3$ in the first place (§3.3), the search found more of the cue the paper had already identified, not a new kind of explanation.

And it is not an artifact of the split the search selected on: frozen, then re-scored on four partitions it never saw. Re-scored on `shuffle_seed` 71, a partition the search never saw, the composite reads $\mathbf{0.7395}$ $[0.6770, 0.7937]$, against the full-token bag's 0.5498, WL_3's 0.2646 and $\phi_{d=4}$'s 0.2641: **the advantage survives and grows**, so it was not fitted to the reported partition. Three further re-scorings extend that to **four** unseen partitions, 71–74 (Table 36), with the six components and all six weights frozen at what the search returned on 70. **The frozen composite stays below the *trained* encoder on every one of them**, by $+0.1635$ to $+0.2670$, and the two class-level intervals are *disjoint* in all five partitions, the encoder's lower limit clears the composite's upper limit by between $+0.07$ and $+0.15$ (stated at two decimals because the largest of the four, 0.1515, sits exactly on the three-decimal rounding boundary). **The margin is narrowest at** 71, the one holdout the paper already had, so the single check it previously reported was the least favourable of the four. The search arm is unchanged work in all four invocations and returns the identical winner and the identical 0.6758, which is how a run that did not re-search is told from one that did.

What licenses reading two runs against each other, and the three things this number is not. The encoder rows come from the five-partition sweep (Appendix AL) and the composite rows from this run, so the join needs a shared object: **both scripts score the same full-token control on the same partition, and they agree**, exactly, to four decimals, at every partition (0.5174, 0.5498, 0.5236, 0.4387, 0.4939), and at 71, where both also record a class bootstrap, the *interval* agrees digit for digit as well ($[0.4609, 0.6289]$ from either script). **That agreement, not an assumption about the two bootstraps, is what permits the non-overlap statement above**; a partition whose shared control disagreed would have been dropped rather than reconciled. Both sides of every row are the sweep's environment, which reads 0.8966 at split 70 where the body prints 0.894, the 0.0023 cross-environment envelope this appendix's companion documents, so no comparison here straddles two environments. *(i)* It is **not a new ceiling**: on 71–74 the composite is a frozen control rather than a search result, and like the full-token bag it does not set $\sup \mathcal{F}_3$, which Track 1 leaves at 0.276. *(ii)* The intervals are **class-level** bootstraps and cover class sampling, not training seeds; Appendix AL bounds that separately and reports trained rows as not bit-reproducible across environments. *(iii)* The composite is the **most split-sensitive object in the comparison** (± 0.056 across 71–74, against the catalogued bag's ± 0.048 and the encoder's ± 0.015), which is itself the argument for reading a ceiling off four splits instead of one. The line the body compares with the encoder stays the one measured on the published split; what these four partitions add is that *removing* the search's oracle access does not change the verdict.

Nor an artifact of $k{=}5$**, the other quantity this cell could have been fitted to.** The ladder records every candidate at each k it sweeps, so the winner's own k-sweep is available without a further run. Against the Tree-LSTM's published sweep at $K{=}500$, $1.000/0.965/0.894/0.710$ at

Table 36: **The searched ceiling, frozen and moved off the split it was fitted to.** The composite was selected with oracle access to partition 70; on 71–74 it is re-scored unchanged and read against the *trained* encoder measured on that same partition by the five-partition sweep (Appendix AL), so both sides of every row come from one environment. *Bag* is the full-token control the two scripts share: its per-partition agreement is what licenses the join, and at 71 its class interval agrees between them digit for digit. **The encoder's interval clears the composite's in all five partitions**, and the margin is smallest at 71, the one holdout the paper already had. Nothing is trained and no search is re-run: all four invocations return the same six-component winner and the same 0.6758 on the search arm. Mean±sd is over the four *unseen* partitions only; Appendix AL's row is over all five.

Split	Search saw it	Tree-LSTM (95% CI)	frozen composite (95% CI)	bag	margin
70	**yes**, oracle	0.8966 [.860, .925]	0.6758 [.610, .735]	0.5174	$+0.2208$
71	no	0.9030 [.866, .931]	0.7395 [.677, .794]	0.5498	$+0.1635$
72	no	0.8750 [.831, .910]	0.6641 [.586, .731]	0.5236	$+0.2109$
73	no	0.8715 [.825, .907]	0.6045 [.529, .673]	0.4387	$+0.2670$
74	no	0.8955 [.856, .926]	0.6562 [.583, .722]	0.4939	$+0.2393$
71–74	mean±sd	0.8862±0.0154	0.6661±0.0556	0.5015±0.0477	$+0.2202$

k=1/3/5/10 (Table 30), the searched composite reads $0.8208/0.7469/0.6758/0.5354$. **The encoder leads it at every** k, and by between $+0.17$ and $+0.22$ throughout: the margin is flat in k rather than largest at the value the body quotes. (We state it as a range because a per-k margin computed from the three-decimal values printed here and one computed from the logs disagree in the last digit; the verifier asserts the bracket against both logs.) Track 1's supremum is not k=5-specific either ($0.2955/0.2841/0.2763/0.2524$). **This does not extend Appendix AI's twenty-cell statement**, which is about that table's catalogued baselines at all five scales: the search ran at K=500 alone, and what it adds is that at *that* scale the ordering survives the k choice against a *searched* baseline and not only against a listed one.

What it costs us, as a number rather than as a framing. §4.2's ratio at K=500 was 0.894 against 0.517, $1.7\times$; against the searched control it is $1.3\times$. Against the encoder's published class-level lower bound 0.8572 the margin falls from $+0.3398$ to $+\mathbf{0.1814}$. **The verdict is unchanged and the margin is narrower**, and the body now prints the *searched* number where it used to print the catalogued one. **This is a result we did not want and went looking for anyway**, which is the only kind that answers the objection it was built for.

What this run does and does not license. It does **not** show $\mathcal{F}_3$ complete, and nothing can: Proposition 3 rules out any finite admissible family licensing the certificate reading, and that is unaffected here. It does **not** widen Definition 1: the composite is order-blind and admissible but is *not* one of the seven, and so, like the full-token and tree-local bags, it does not set $\sup \mathcal{F}_3$, which Track 1 leaves at 0.276. **What it does establish is the one thing a declared family cannot establish by declaration**: the specific failure the objection describes, a stronger order-blind explanation the list happened to omit, was searched for adversarially, with oracle access and a composing search space, and the strongest one found still leaves the encoder's class-level interval clear of it by 0.18. **And it establishes something against us**: our published catalogue understated the strongest available order-blind control by 0.16, so the honest reading is that a *list* of controls is a weaker instrument than a *search* over them, and a paper that reports the list alone is quoting a flattering ceiling. **But no published number of ours was *wrong***: what was too weak was the *instrument*, a list where a search was available, so this is not an eighth entry in Table 11's ledger of defects, which is what that table counts, but a narrowing of our own margin, and it is recorded as one in Table 10. **The body carries the searched number.**

Reproduction. `python3 run_r101_adversarial_family.py` ($\approx$ 10.4 min $=$ 626.6 s, CPU only; nothing trained, nothing downloaded). Search seed 101, `shuffle_seed` 70 for the search and 71 for the holdout, k=5 with $k \in \{1, 3, 5, 10\}$ recorded, `twin_K`=500. The three further holdout partitions are the same command at `--holdout_seed` 72, 73, 74 under tags `r101_adversarial_family_h72/h73/h74`, which is why the shipped log is not overwritten. **The pre-committed interpretation (all four outcome branches, including the one that would have bounded our own claim) is in the script's docstring, written before the run and not renegotiated after**, as `r92`'s is; the four branches for the added par-

titions, including the one under which the body would have had to narrow the claim, are in `PREREGISTRATION_r101_holdout_sweep.md`, likewise written first.

BC. Composition Novelty as a Lattice: an Unseen *Triple* Costs Nothing, and the One Cost We Can Resolve Sits on `commute` (tag `r102_composition_lattice`)

Why this run exists, and why it is the only run this revision added. Two consecutive reviews asked for no further experiments. Six of the seven conditions the seventh asks be controlled separately already have tags, primitives in Appendix AQ, depth and adjacency in Appendix AU, ordering in Appendix AV, rewrite families in Appendix AP, a second algebra in Appendix AO, and **one did not exist**: an unseen *triple*. Appendix AU's own disclosure (iii) had said so in print (*"three bigrams, not twelve… all three involve `commute`"*), which names two limits at once, the n-gram order and the fact that every withheld combination contained the one $\delta{=}0$ primitive. **This run closes both**, and it is reported here rather than in the body because what it establishes is a *bound on what the composition result licenses*, not a new capability.

Why co-occurrence and not adjacency, which is the design that would have been decided before it ran. The obvious construction is Appendix AU's, one order up: withhold an adjacent *trigram* instead of an adjacent bigram. **That contrast is forced by the sample space**: fewer depth-8 orders place three named primitives contiguously than place two, so withholding the trigram constrains training strictly *less* than withholding the bigram, and $\mathrm{cost}(\text{triple}) \leq \mathrm{cost}(\text{pair})$ would have held whatever the encoder learned. So the manipulated variable here is **co-occurrence**, which *combinations* of known primitives appeared in the training library, over which order $1 \subset 2 \subset 3$ is a genuine lattice. **Arrangement is held fixed**: a type's training composite and its test form are composed in the *same* per-class drawn order, so arrangement remains Appendices AU and AV's axis and is not re-measured here.

Eight arms, matched in library size and in token budget rather than argued to be. `poly8`, $K{=}200$ classes, the four primitives at token deltas `commute` 0, `double_negate` 8, `add_identity` 4, `mul_identity` 4. The reference arm COVERED-PAIR trains the four singles and all six pairs: 10 **forms, total token delta** 64. Each of three NO-PAIR arms trains the same 10 forms with one pair replaced by a per-class length-matched composite of the *same* four primitives that never composes the withheld combination (`mul_identity+mul_identity` for 41 of the 200 classes on the first, for instance), so 10 **forms and delta** 64 **again** and what is manipulated is the specific co-occurrence, not composition and not library size. COVERED-TRIPLE adds all four triples: 14 forms, delta 112; each of two NO-TRIPLE arms replaces one triple the same way, 14 and 112. SIZE-CONTROL matches 14 and 112 with four filler forms each spanning *at most two* distinct primitives, so it pays the covered-triple library's exact size and token cost while training **no three distinct primitives together at all**. Three of the six pair types and two of the four triple types are withheld; **two of the three withheld pairs and one of the two withheld triples contain no `commute`**, which is the half of disclosure (iii) a wider sweep alone would not have closed. `training_chains_composing_the_withheld_type` is recomputed per arm and reads 0 in all five withholding arms, and a leak check over all eight arms reports 0 **coincidences**.

What certifies that a test form really *requires* the withheld co-occurrence. A class is admitted only if, for every one of its five test forms, a route search at radius 3 (24 samples, frontier cap 1600, mean route map 5592.5 forms) **fails to find any derivation of that form which avoids the co-occurrence**. The test is one-sided by construction (a class is rejected on a route *found*, never on a search that came up empty) so it needs a power statement and a control, and both are logged: the search reaches 926 of 1000 sampled test forms (power 0.926, with 200 searches hitting the frontier cap), and **the same search asked about a co-occurrence the form is known to contain fires on** 923 **of** 1000, so it does find an avoiding route when one exists. 209 classes were scanned to fill 200; the 9 dropped all failed on `test_forms`. The class set is built once and *fixed* across training seeds, so no arm withholds from a different set of classes.

The costs, paired per class on the byte-identical test form. Every number in Table 37 is a difference between two arms scored on the *same* held-out form of the *same* class, so a form-level cue differences out exactly and no shape-matched twin is needed (Appendix AU needs one because

Table 37: **Takeaway: withholding a *triple* costs nothing, withholding three-way training entirely costs nothing, and the only interval that clears zero does so by one step of the statistic's resolution, on the one withheld pair containing `commute`.** Primitives abbreviated c = `commute` (δ=0), n = `double_negate` (8), a = `add_identity` (4), m = `mul_identity` (4). Each cost is a per-class paired difference against the arm that trained the combination, on the byte-identical held-out form; 95% intervals are class-level bootstraps over K=200, and the statistic's grid is 0.001. *Resolves* means the interval excludes zero. The library-size rows are the confound test: none of them resolves, and the largest is 0.005, so no cost above is attributable to the extra forms.

Withheld from the training library	order	δ	paired cost (95% CI)	resolves
`c+n`	2	0+8	$+0.010$ $[+0.001, +0.021]$	**yes**
`a+m`	2	4+4	-0.002 $[-0.006, +0.000]$	no
`a+n`	2	4+8	$+0.000$ $[-0.003, +0.003]$	no
`a+c+m`	3	4+0+4	-0.002 $[-0.005, +0.000]$	no
`a+n+m`	3	4+8+4	-0.001 $[-0.005, +0.002]$	no
all three-way training, tested on `a+c+m`			$+0.000$ $[-0.003, +0.003]$	no
all three-way training, tested on `a+n+m`			-0.002 $[-0.005, +0.000]$	no
library size alone, on the three pair tests			$+0.005$, $+0.003$, $+0.001$	no
three-way training's effect on the pair tests			$+0.004$, $+0.004$, $+0.001$	no

it reports absolute accuracy on a single arm). **The class-mean paired gap moves in steps of exactly** 0.001 (five seeds of binary correctness per class, averaged over 200 classes), which is the resolution the next paragraph turns on.

The pre-registered branch fired, and its own label is wrong, which is a defect in our pre-registration, found by executing it. Four branches were written into the runner's docstring before the first epoch and the branch is *computed in code*, not chosen afterwards. It returned **(3), "co-occurrence costs even at order 2"**, and it returned it because one of the five intervals excludes zero. **That label is not what the magnitudes say.** The mean unseen-pair cost is $+0.0027$ and the mean unseen-triple cost is -0.0015, the two figures on this page that are legitimately off the grid, being averages of three and of two on-grid values, while the single resolvable cost is $+0.010$, whose interval clears zero by $+0.001$, **exactly one step of the statistic's grid**; and §4.5's own axis table prints $+0.012$, a *larger* number, in the row it labels *free*. **The predicate was keyed on statistical resolution with no smallest effect size of interest**, and at K=200 against a reference arm above 0.99 the bootstrap resolves a third of a point. So the branch is reported as it computed, beside the magnitude that contradicts its wording, and the lesson is recorded rather than repaired after the fact: **a pre-registered decision rule that tests only whether an interval clears zero will, at this sample size and this ceiling, label as a cost an effect the same paper calls free four pages earlier.** The confound test is clean (no library-size interval clears zero, the largest is 0.005) so `confounded_by_library_size` is `False` and nothing here is attributable to arm size.

What the numbers do license, and where the cost actually sits. Three statements survive the paragraph above. *(i)* **Composition novelty is bounded at** $+0.010$ **at order** 2 **and is not resolvable at order** 3: both triple costs are *negative* point estimates with intervals covering zero, which at this ceiling is noise and not a benefit. *(ii)* **Withholding three-way training altogether costs nothing measurable** on either triple test, against a size- and token-matched control, a stronger withholding than the reviewer asked for, and the arm that never saw `a+c+m` reads 1.0000 on all five test sets in all five seeds. *(iii)* **The one resolvable cost is the only withheld pair that contains `commute`**, and *both* `commute`-free withholdings cost nothing, so disclosure (iii)'s worry that the adjacency result rode on `commute` was not a sampling accident but a locus: the cost lives on a primitive, at order 2, and does not escalate with order. That is the same direction as Appendix AQ's leave-one-primitive-out result and the same reading the review states in its own words, that the difficult part is acquiring the primitive invariances rather than composing them.

What it does not license, starting with the ceiling. Every arm scores between 0.985 and 1.000 on every test, so **this design bounds a cost and cannot resolve one**: an effect below about 0.01 is beyond it, and nothing here speaks to a regime at lower absolute accuracy. It is not a trivial test, the strongest of 14 zero-parameter bags reaches 0.115–0.260 across the five test sets and an untrained

encoder of the same architecture 0.257–0.337, against chance 0.005, so by the paper's own instrument this is an arm the audit *admits* and the encoder clears by a wide margin, which is exactly why the ceiling is a property of the encoder here and not of an easy split. And it does not show that a general compositional operator was learned: one corpus, four primitives, one architecture, and the property tested is recovering a held-out form's class. **The body's §4.5 row is annotated with the bound, not rewritten**, and Figure 3's arrangement bar and its provenance pin are untouched, because arrangement and co-occurrence are different manipulations and this run measures the second.

Reproduction. `python3 run_r102_composition_lattice.py --tag r102_composition_lattice` (10702.9 s $\approx$ 2.97 h on MPS; 8 arms $\times$ 5 seeds $=$ 40 trainings, plus construction, certification and the zero-parameter ladder). `arch=tree`, `embed_dim` 128, 10 epochs, batch 64, `lr` 10^{-3}, margin 0.2, `K` 200, `build_seed` 0, `route_radius` 3, `bf_max_len` 3; the tag was checked absent from `logs/` before launching, because `--tag` names both the JSON and the `.log`. **The four outcome branches, the confound test and the reason adjacency was rejected as degenerate are in the script's docstring, written before the run**, and are emitted verbatim into the log as `preregistered_branches`; `--build_only` reruns construction, matching, certification and every assertion without training.

BD. How Much Larger Can the Rewrite Library Get? The Library Ceiling, and Why the Depth-8 Size Contrast Was Not Trained (tag `r103_library_size`)

Why this section is a census and not an accuracy table. The request is for a larger transformation library. **We measured how large one can be before designing anything, and the answer is five**, on `poly8`, the corpus the paper already uses. So the paper's most-quoted concession, that four primitives are not a library, is not an admission of insufficient effort: it is one short of a ceiling set by the released data. Two censuses establish this, neither needing an encoder, and the second of them then *stopped* the contrast this section was written to run. Both were taken before the design was fixed, and the pre-registration is `PREREGISTRATION_r103_library_size.md`.

Census 1: every corpus the survey covers, every rewrite schema this project ships. Three properties per (corpus, schema), each thresholded at 20% of that corpus's anchors: *usable*, the schema has at least one site; *inventory-safe*, every operator the rewrite *introduces* already occurs in that corpus; *repeatable*, it is inventory-safe *and* has two or more distinct sites, so it can appear twice in one chain. **Inventory-safety is a tier condition, not a technicality**: a rewrite that introduces a new operator makes the held-out form separable by a token multiset, which is an **S1** cue by this paper's own definitions (§3.2), so a library that grows by such a schema does not measure composition at all. The method reproduces `run_r91_nonlocal_composition.py`'s own recorded site count for `distribute` on `poly8` exactly (283 anchors with one site, 0 with two).

The sixth usable schema is disqualified by this paper's own framework, and it had never been measured. `_rw_power_to_explog` ($a{*}a \to \exp(2\log a)$) appears nowhere in any earlier draft of this paper, and it fires on 313 of `poly8`'s 1102 anchors with a constant token delta of $+6$ on all 316 leaf sites, mechanically the cleanest of the thirteen. **And it is inadmissible**: `poly8`'s entire operator inventory is $\{+, -, *\}$, so the rewrite introduces exp and log, which occur *nowhere* in the corpus, and a held-out form carrying it is separable from every distractor by the single question *does the token multiset contain* exp. Adding it would have raised the library to six and measured an S1 cue. We report it because a schema the corpus admits and the paper never measured is a gap a reader can find, and because it is the reason the ceiling is five rather than six. The remaining seven poly-side schemas, `reassoc`, `explog_to_power`, `reciprocal_to_explog`, `exp_log_inverse`, `log_exp_inverse`, `remove_add_identity`, `remove_mul_identity`, fire on **zero** of 1102 anchors. The last two are instructive: EQNET's forms are already minimal, so an *inverse* rewrite has no site until a forward partner creates one, at which point the pair cancels and a nominal depth-8 chain is really depth 6. That design was ruled out before it was coded.

Census 2: at five members, depth 8 is barely reachable, and that is structural, not a search budget. The composition result is at depth 8 with *delta-matched* steps: every rewrite contributes its canonical token delta, so a held-out form cannot be identified by length (Appendix AU). Per anchor we attempt a depth-8 delta-matched chain and record the depth reached. Over the paper's four

Table 38: **Takeaway: the largest inventory-preserving rewrite library available anywhere in the survey is five, and the paper's is four.** A census over every corpus EQNET releases and all 13 poly-side plus 4 boolean-side schemas this project ships; no encoder, no training. *Usable* counts schemas firing on at least 20% of a corpus's anchors, *inventory-safe* those introducing no operator absent from that corpus, *repeatable* those *inventory-safe* schemas with two or more distinct sites on one anchor, nested, because a schema that cannot enter the library cannot repeat inside a chain, and the two readings differ only on `simplepoly`, where `mul_identity` repeats but introduces the $*$ those corpora do not contain. The boolean 4 is a code ceiling (`bool_rewrites.py` ships four schemas in total), not a firing rate; †five of the eight boolean corpora also carry `xor` and `implies`, which no boolean schema introduces, and the 3 is `boolean5` and `largeBoolean5`, where `de_morgan` has a single site. `poly8`, in bold, is the corpus every composition result in this paper uses. **The repeatable column is why depth 8 is reachable at four members and barely reachable at five**: a chain of eight delta-matched steps over a five-member library needs three repeats, and nowhere does a fifth member repeat.

corpus	classes	inventory	usable	inventory-safe	repeatable
`simplepoly5` / `8` / `10`	47 / 104 / 195	`+ -`	4	3	3
`poly5`	150	`* + -`	5	4	4
`poly8`	**1102**	`* + -`	**6**	**5**	**4**
`oneVarPoly10` / `oneVarPoly13`	83 / 677	`* + -`	6	5	4
all eight boolean corpora†	95–7312	`and or not`	4	4	3–4

primitives, each allowed twice, 987 **of** 1102 **anchors reach depth** 8 **(**89.6%). Over the inventory-safe five, with `distribute` exactly once, 175 **of** 1102 **do (**15.9%). **Raising the search budget does not recover it**: at `step_tries` of 200 / 600 / 2000 the five-member count reads 175 / 212 / 198 and the four-member count 987 / 1002 / 989, a plateau near 19%, so the variation is which chains the shared generator draws and not how hard it tried. The cause is in the repeatable column of Table 38: `distribute` has two or more sites on 0 **of** 1102 anchors, so it can never repeat, and its single site must survive the seven rewrites applied around it while staying at its delta-+4 sites. Appendix AS hit the same wall at depth *four*: 1102 anchors scanned, 133 retained, largest drop reason `depth_ladder`.

The contrast that was designed, and what makes it a contrast rather than two runs. Two arms at fixed data quantity and fixed held-out form: L4, the paper's four primitives each allowed twice, and L5, the inventory-safe five with `distribute` exactly once. **Both arms are built in a single pass over one anchor, so the paired class set is an intersection by construction rather than an intersection computed afterwards**, a class enters only if *both* arms admit it, and the two arms then share the depth-8 ladder and its per-depth shape-matched twin byte for byte. Library sizes are equal within an arm, the L5 composite chain is *searched* for the same total token delta as the L4 chain rather than padded to it (Appendix BC's discipline), and the closure guard is searched with the full five-member set so a route that avoids `distribute` disqualifies the class in *both* arms. **This is not a leave-one-primitive-out test**; that is Appendix AQ's experiment, and it is what putting `distribute` in the held-out chain would have measured instead. All nine construction guarantees are computed and logged as booleans, not asserted in prose, and all nine read `true`.

The stop rule fired, and it fired at both search budgets, including the one census 2 was measured at. The rule was registered before the run: **fewer than** 120 **paired classes means do not train**, with K requested at 150. The first invocation retained 89 **of** 1081 **anchors scanned**. Its search budget (`step_tries` 12, `site_tries` 30) was, however, far below the 200/600/2000 census 2 reports, so **a feasibility stop rule firing there would have been an artifact of the run's own budget**, and the run was repeated at the census's setting, `step_tries` 200 and `site_tries` 60. It retained 97, of the same 1081. **The drop table is what settles it, not the retention count.** Between the two invocations `depth_ladder`, the one bucket a larger step budget can empty, *fell* from 55 to 42, while `no_distribute_site`, the structural one, *rose* from 662 to 682; the rest read 114 → 108 (`no_site_for_a_primitive`), 112 → 108 (`no_twin`), 47 → 39 (`l5_composites`) and 2 → 5 (`composites`). So 16.7× the step budget and 2× the site budget bought the eight classes a budget can buy and left the constraint that matters where it was: **two thirds of `poly8` has no delta-matched `distribute` site at all**, which is Table 38's repeatable column again. Each invo-

cation took about 18 minutes and *trained nothing*, both are `--dry_run`, and all nine construction guarantees read `true` in both, so the STOP is a statement about class supply and not about a build that failed to assemble.

The projection that justified requesting $K{=}150$ was itself wrong, by about a factor of two. The pre-registered rationale projected $\sim$186 paired classes, `distribute`'s 283 admitting anchors times the 0.73 construction factor Appendix AU measured at four primitives, which is why 150 was requested and why a stop rule was needed rather than assumed. Measured: 89, then 97. **The projection multiplied two retention rates as though they were independent, and they are not**: an anchor that admits a delta-matched `distribute` site must *also* admit a full eight-step delta-matched ladder built around that site while leaving it intact, and conditioning on the first makes the second markedly less likely. This is not counted among the four failed registered predictions §1 reports; those are point predictions and pre-committed branches, each asserted from its own log in the failing direction, but it is the reason the rule was written, and it is printed rather than repaired.

What the two censuses license, and what they do not. No accuracy claim is made in either direction. We do not report that a five-member library helps, hurts or leaves the depth-8 result unchanged, because no encoder was trained at $|L|{=}5$ on a paired class set that clears the registered floor, and reporting a contrast from 89 classes after registering 120 would be the practice this paper spends nine pages auditing. What the censuses do license is narrower and does not need an encoder: **the largest inventory-preserving rewrite library available anywhere in EQNET's released corpora is five, the paper's is four, and depth-8 delta-matched composition over five is reachable on 15.9% of `poly8`'s anchors against 89.6% over four. The scope of that is exactly 15 released corpora and the 17 rewrite schemas this project ships**; it does *not* say a larger library is impossible, and a corpus with a fifth schema that has two or more distinct sites per anchor would supply one immediately. That, not more classes, is the missing resource: the ceiling is set by *repeatability*, not by corpus size, which is why `oneVarPoly13`'s 677 classes and `poly8`'s 1102 return the same five. So the paper's closing concession is neither softened nor withdrawn here (four primitives are not a library); it is *explained*: one short of a ceiling we measured before designing anything, and the design that would have reached that ceiling stopped on its own rule.

Reproduction. `python3 run_r103_library_size.py --dry_run` reproduces the first invocation (1107.9 s; defaults K 150, `depth` 8, `k_para` 6, `n_comp` 2, `site_tries` 30, `step_tries` 12, `seed` 0, `corpus` `poly8`, `min_classes` 120, tag `r103_library_size`), and `--tag r103_library_size_s200 --dry_run --step_tries 200 --site_tries 60` the second (1099.6 s). `--tag` names both the JSON and the `.log`, so the second invocation carries an explicit distinct tag rather than overwriting the first. The pre-registration, including the stop rule, both smallest effect sizes of interest and the kill clause that the S2 operator bag would have triggered, is `PREREGISTRATION_r103_library_size.md`, written before either invocation. `python3 census_r103_library.py` reproduces Table 38 over all 15 corpora without training anything.

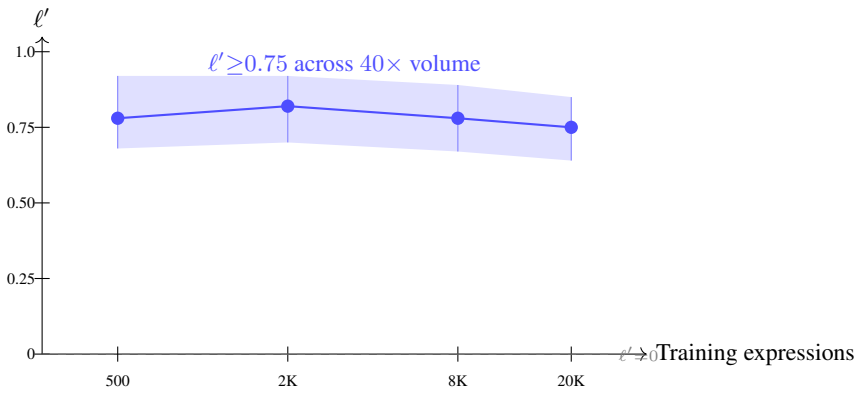

Figure 6: Augmentation-volume robustness (E3b protocol, Tree-LSTM, n=10/5/5/25 seeds at 500/2K/8K/20K respectively, bootstrap 95% CIs). ℓ' remains high ($\geq$0.75) at every volume tested. The 20K point derives from tag `r34`; 500–8K points from `r39b/d/f`. Between-run-family differences ($\sim$0.10) exceed any volume effect, so we report levels rather than a fitted trend. More gradient exposure to paraphrases of the same equations does not teach variable-identity invariance.