GRAPHS ARE EVIDENCE, NOT CIRCUITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph learning has mistaken its evidence for its machinery. The evidence says what is observed; the query says what must be inferred; the circuit specifies how observations are combined. Message passing commonly inherits that circuit from the observed graph. We separate these choices and show why the distinction matters. Changing only the query on identical evidence moves a linear predictor from 58 accuracy points below an exact solver to within 0.1 point of it. Recompiling Minesweeper's clue relationships improves a parameter-matched GCN by 5.24 AUC points; randomizing those relationships erases the gain. Exact examples explain why seeing farther can still leave a linear predictor below optimal performance. **Two-System Graph Learning (TSGL)** puts the separation to work: preserve a strong GNN, learn explicit local relationships within a declared library, and add inference only when held-out validation shows a gain. Without label inputs, it reaches state-of-the-art-level Minesweeper performance at 99.09 ROC-AUC; training-label inputs raise this to 99.97. Relative to its matched baselines, remaining ranking error falls by approximately 63% and 98%. On arxiv-year, the gain persists beyond generic correction and a second GNN under matched label access. A separate model learns numerical relationships within a supplied counting family and reuses them on independent boards without further training. For questions about groups of cells, it improves on treating its individual predictions as independent. Treat the graph as evidence, then build the computation the question requires.

1 INTRODUCTION

Consider the two Minesweeper clues in Figure 1. What do they tell us about the unseen cell x_1? One says that x_1 and x_2 contain one mine; the other says that x_2 and x_3 contain two. The second clue fixes $x_2 = x_3 = 1$. Substituting into the first gives $x_1 = 0$. We have answered the question by composing two observations. The grid identifies neighbours. The deduction uses a different object: a sequence of operations on overlapping constraints.

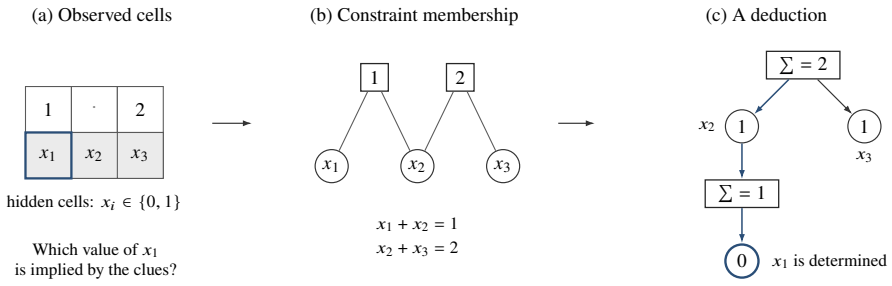


Figure 1: **The observation graph does not specify the deduction.** Panel (b) groups the variables in each clue; panel (c) composes the constraints to determine x_1. No new observation is added. Only the displayed clues are used; the known-empty dot's clue is omitted.

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.

Message-passing graph neural networks (GNNs) commonly use their input graph twice: once as the carrier of evidence, and once as the circuit that assembles an answer (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018). This identifies two different objects. Observed edges describe relationships in the data; computational dependencies specify how to combine observations for a query. Their identification is an inductive bias, not a requirement of graph learning. Even a graph irrelevant to the target can influence a GNN (Bechler-Speicher et al., 2024).

The central claim is that *graph learning must distinguish the structure of observations from the structure of inference*. We establish this through three contributions. *A formulation:* evidence, query and circuit are separately variable parts of the learning problem. *A controlled analysis:* changing only the target on one evidence array isolates query requirements; recompilation under a declared GCN budget tests circuit value; factorial interventions separate membership, operators and anchor access. Exact ranking constructions explain a reach–interaction separation for fixed linear propagation. *Two constructive consequences:* TSGL tests whether relational computation improves a capable predictor, while a count-family compiler tests whether learned relationships remain useful for new instances and joint queries.

Minesweeper is a flagship consequence: near-perfect prediction, recovery of exercised counting relationships, and reusable joint inference. Relative to its access-matched baselines, TSGL removes approximately 63% of remaining ranking error without label inputs and 98% with training-label anchors. Arxiv-year gains exceed generic correction and ensembling. Together, these results connect controlled separation to practical prediction and learned-object reuse.

2 Separate the evidence, the query and the circuit

In brief. The observed graph is held fixed while the query and circuit reveal distinct determinants of performance.

Let $\mathcal{E} = (G, X, O, M)$ collect the observed evidence: graph, features, revealed values and observation masks. A query q specifies a target y_q; an inference circuit $\mathcal{C}_q$ specifies intermediate quantities, dependencies and update operations:

$$\underbrace{\widehat{y}_q}_{\text{answer}} = \text{Execute}\big(\underbrace{\mathcal{C}_q}_{\text{circuit}}; \underbrace{\mathcal{E}}_{\text{evidence}}, \underbrace{\theta}_{\text{parameters}}\big). \tag{1}$$

A circuit may reuse adjacency, introduce factor nodes, or follow a trellis. Equation (1) makes that choice explicit rather than inheriting it from observed connectivity. The task is the tuple *(generator, observation model, query, training access)*, not the graph alone. An *anchor* is a revealed target value that inference may use.

2.1 Keep the evidence; change the question

A finite-state machine (FSM) evolves as $s_{t+1} = \delta(s_t, x_t)$. Each of 20,000 time positions is a node of an undirected path, carrying a noisy observation of symbol x_t. At a shared 2% anchor mask we additionally reveal both s_t and x_t. The probe propagates this same feature array over the path for both tasks, with identical split masks. Only the target changes: recover a state or a true symbol. A forward–backward reference receives the true transition table and noise law (Appendix E.3).

On the first draw, state accuracy is .2017 for the linear probe and .7827 for the exact trellis reference. Symbol accuracy is .9512 and .9515, respectively: a 58.07-point change in reference deficit. The contrast repeats on three additional draws and a new state-space size (Figure 2). The same evidence supports a near-reference linear solution to one question and a large deficit on the other. The intervention isolates what is asked, not how much is observed.

2.2 Keep the question; change the circuit

On Minesweeper (Platonov et al., 2023), we retain a residual GCN update family, observations, supervision and training recipe. Parameter counts match within 0.7%. At eight layers, compiling revealed clues into variable–factor membership raises AUC from .9035 to .9437 ($+4.02$ points). Validation selection from the preregistered 8/16-layer compiled menu yields .9559 ($+5.24$ points; Figure 3).

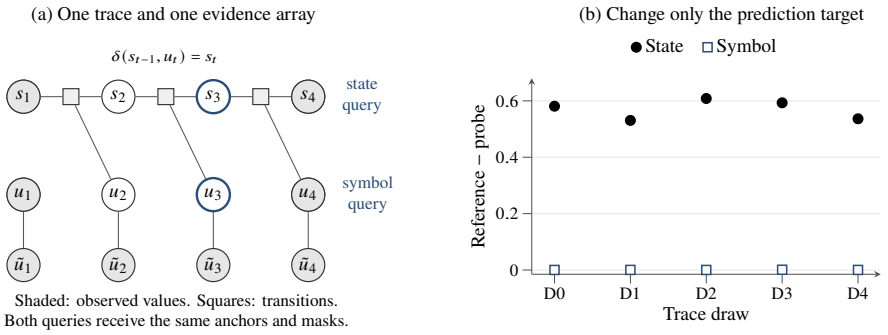


Figure 2: **Same observations, different questions.** Both arms receive all shaded observations. Across five draws, exact-reference minus probe accuracy is .531–.608 for states and at most .001 for symbols. D0–D3 use eight states and four symbols; D4 uses sixteen states. Zero denotes no reference gap, not chance accuracy.

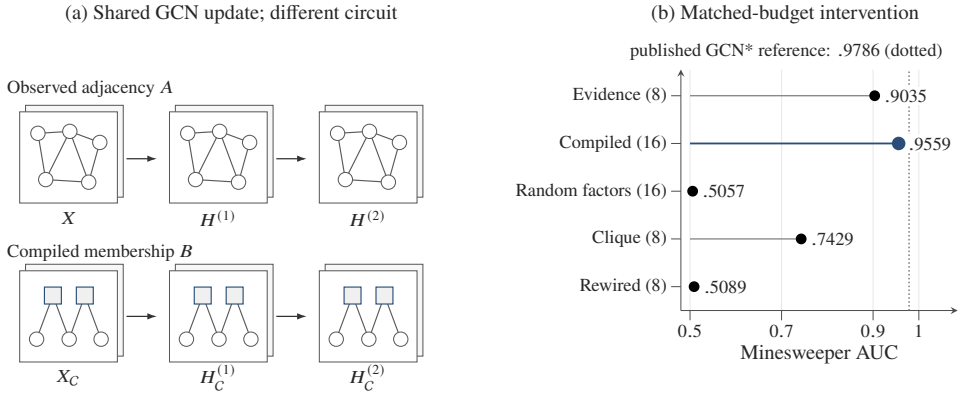


Figure 3: **Recompilation improves a common GCN update family.** Left: observed edges versus factor membership. Right: parameter-budget-matched arms; the compiled depth is validation-selected (16 versus 8 for evidence adjacency). The eight-layer compiled arm also gains 4.02 points. Dotted line: separately published tuned GCN* (Luo et al., 2024).

The depth menu accounts for two-edge variable–factor–variable communication. The bipartite input has separate type encoders (Appendix E.1). Random membership with the same sixteen-layer model gives .5057; clique completion gives .7429 and rewiring .5089. These controls do not reproduce the membership gain. Cora changes by -3.3 points under replacement and $+.25$ under augmentation (Appendix E), establishing task dependence.

2.3 Membership, operators and anchors interact

A preregistered $2 \times 2 \times 2$ design varies true/random factor membership, typed/affine kernels and clamped/absent anchors. Minesweeper uses five official splits; the FSM replication uses five draws and a fixed BP solver. With anchors, the typed-kernel gain is $+10.94$ AUC points under true Minesweeper membership but -1.64 under random membership. The FSM gains are $+63.49$ and $+.49$ accuracy points. The large clamped benefit depends on matching the operation to the correct members.

Correct observations can amplify a wrong circuit. Clamping anchors into randomized typed Minesweeper factors reduces AUC from .7033 to .4992: -20.41 points. The preregistered worst-cell ordering was wrong, a scored miss. The negative anchor effect was then predicted before the FSM replication and held there (Appendix F). Membership, operations and observation access must be evaluated together.

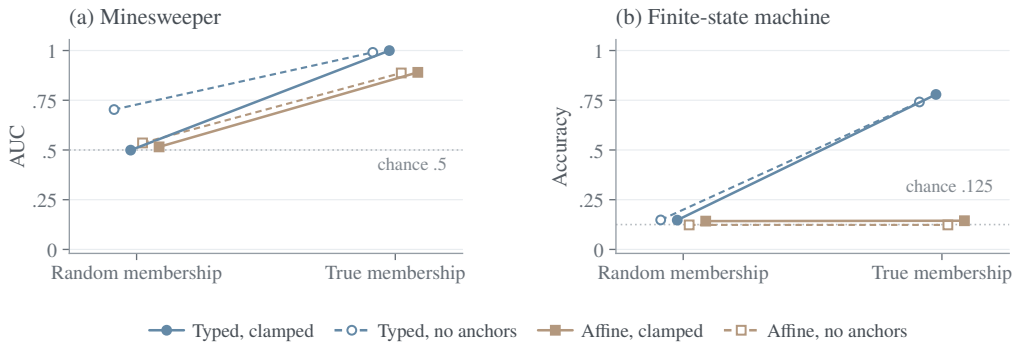


Figure 4: **Typed operators help when connected to the right variables.** Blue circles denote typed kernels; sand-colored squares denote affine kernels. Solid lines and filled markers clamp anchors; dashed lines and hollow markers omit them. Dotted lines mark chance. All sixteen means are shown, with small horizontal offsets for legibility. Randomizing membership removes the large clamped typed-kernel advantage. Full means and frozen misses: Appendices E and F.

3 Reach is not interaction: an exact separation

In brief. Radius changes which observations a linear score reaches, not the algebraic interactions it can express.

Let observation signs be $\sigma \in \{-1, +1\}^K$, with affine encoding $X(\sigma)$. For a linear propagation operator P fixed with respect to the evaluated observation, set

$$F_r = [X, PX, \ldots, P^r X], \qquad S_i(\sigma) = w^\top F_r[i] + \beta. \tag{2}$$

Theorem 1 (Affine confinement)**.** *Every score in Equation (2) is affine in σ, at every radius. Concatenating multiple fixed linear operators or fixed affine preprocessing does not change this conclusion.*

Linear maps preserve affine functions: increasing radius or changing fixed connectivity cannot introduce a missing product into this score class. A fixed strictly increasing logistic link preserves its binary ranking. The decisive object is the posterior *ordering*, not merely a nonlinear posterior; the next result gives sufficient conditions for a strict AUC separation.

Theorem 2 (Sufficient ranking obstructions)**.** *On a finite sign cube with positive class-conditional atom masses, the best affine AUC is strictly below Bayes if either (a) the likelihood ratio has unique highest and lowest atoms that are not antipodes, or (b) both class-conditional distributions are sign-flip invariant and their likelihood ratio is nonconstant. In (b), every affine score has AUC $1/2$.*

An affine score's unique extrema are antipodal. Under sign-flip symmetry, its class-conditional scores are symmetric about the same center. These observations give the ranking obstructions; finiteness makes the gap strict uniformly over affine orderings (Appendix A).

Table 1: **The query determines the affine gap on the same evidence.** Rows 1–2 use independent Bernoulli(p) bits and $0 < p < 1/2$. The balanced noisy-parity model in row 3 is defined in Appendix A, with $0 < \varepsilon < 1/2$. Entries are population-optimal AUCs in the stated classes.

Construction / query	Bayes AUC	Best affine AUC	Gap
Two checks / leaf	$(1-p)^2(1+2p)$	$1-p$	$p(1-p)(1-2p)$
Same checks / center	$(1-p)^2(1+2p)$	Same as Bayes	0
Noisy parity	$1-2\varepsilon(1-\varepsilon)$	$1/2$	$2(\varepsilon-1/2)^2$

For independent bits i, j, k, observe $\sigma_a = (-1)^{i+j}$ and $\sigma_b = (-1)^{j+k}$. With $L_0 = \log((1-p)/p)$, the leaf's posterior log-odds are $L_i = L_0(1+\sigma_a+\sigma_a\sigma_b)$; the center's are $L_j = L_0(1+\sigma_a+\sigma_b)$. The former has a strict affine gap, .04275 at $p = .05$; the latter has none (Table 1). Identical observations, different query, different algebraic requirement: this is an exact counterpart to the empirical query intervention.

From expressivity to learning. The separation concerns fixed linear propagation and affine readout. Nonlinear GNNs, transformers and observation-dependent operators can create richer interactions (Xu et al., 2019; Rampášek et al., 2022); learning the useful ones under a given access budget is an empirical question. The theorem identifies a resource that reach alone cannot provide in its score class; the GCN study tests explicit circuit construction within a declared nonlinear-model budget.

Diagnose reach and interaction separately. Linear probes measure what radius makes accessible to a linear readout; nonlinear heads test additional interactions, and law-aligned references measure headroom. These are readout-specific diagnostics, not universal ceilings (Appendices D and F). TSGL selects candidates empirically rather than deriving deployment decisions from the probe.

4 Two-System Graph Learning

In brief. Preserve strong perception and add the computation that improves it.

TSGL separates perception, structured inference and acceptance (Figure 5). Validation nodes are divided into a selection set V_{sel} and an acceptance set V_{acc}. The system retains a saved baseline p_0, fits specialist candidates on permitted raw evidence, and selects a probability mixture on V_{sel}:

$$p_\lambda = (1-\lambda)p_0 + \lambda p_s, \qquad \lambda \in \{.25, .5, .75, 1\}. \tag{3}$$

The design objective is *complementarity*: learn corrections that improve p_0, with the preserved predictor itself a first-class execution path. Without acceptance, TSGL returns p_0 pointwise. Both access protocols fit training targets: *strict* forbids label inputs; *anchored* also permits training labels as inputs.

Perception and evidence. The perceiver is selected from declared GCN, GraphSAGE, APPNP or directed-SAGE recipes. Specialists receive original and fixed propagated features, preserving incoming/outgoing roles on directed graphs. Raw evidence remains accessible; baseline probabilities enter the output mixture directly. Training, checkpoint/weight selection, acceptance and test nodes are disjoint. Appendix B gives masks and menus.

The relational branch. For at most ten classes, the engine uses learned affine, cardinality and parity updates, ten damped rounds, and feature-affinity membership capped at eight neighbours per role. One membership policy was chosen on development tasks and frozen; ties are broken by node ID, making membership invariant to edge-storage order. All-edge feature propagation remains available; explicit higher-order factors use the retained members. Larger class counts use an uncapped pairwise approximation. Within this declared library and membership policy, the learner fits local tables and type mixtures. Two generic correction heads receive the same feature bank and comparable selection access, isolating value beyond an additional prediction head.

Isolated acceptance and exact preservation. A saved proposal is accepted only when its acceptance gain is at least .002 and exceeds c paired standard errors: $c = 2.24$ in the ten-task panel and 2.50 in the later arxiv comparison. Accuracy uses paired correctness differences; AUC uses paired positive/negative placement differences. The preservation identity is exact, without a sampling assumption. Acceptance on dependent graph observations is an empirical decision rule, not a guarantee of future no-harm. Appendix B gives the computation; Appendix A distinguishes the older IID argmax analysis.

The anchored baseline is selection-best among the original predictor, a label-input adaptation and Correct&Smooth (Huang et al., 2021). Label-input fitting masks a random half of training labels as observations and predicts the remainder; acceptance/test labels never become inputs. Anchored gains are therefore measured against a baseline that can use the same label observations.

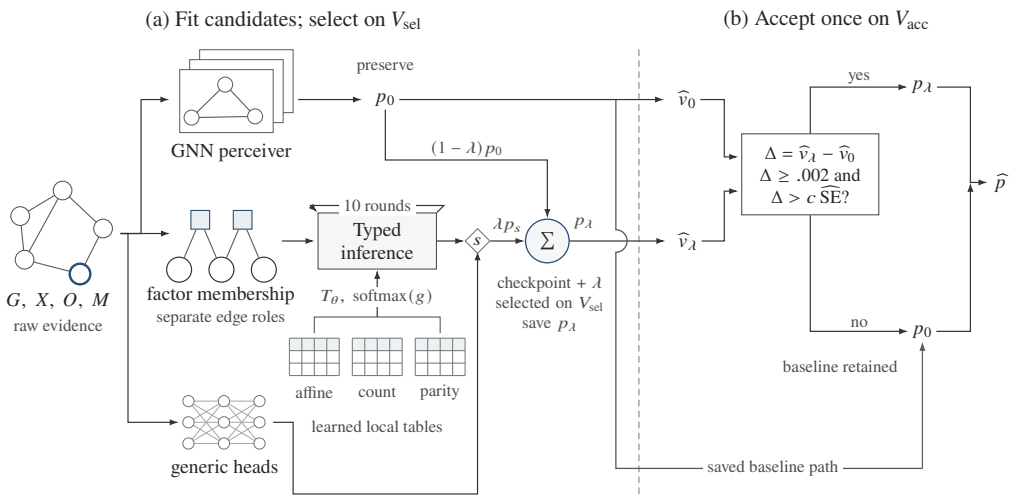


Figure 5: **Preserved perception with learned relational supplementation.** Fit tables T_θ and type gates g on training targets; choose the checkpoint, candidate and mixture weight on V_{sel}. Lock the proposal before V_{acc}. Deploy the accepted mixture or retain the saved baseline pointwise, without executing the specialist. The later arxiv control adds an independently fitted second GNN.

Table 2: **Separate the learning problem from the information supplied.** Training targets, label observations and structural assumptions are distinct resources. Each branch below has its own explicit access contract.

Branch	Target-label fitting	Label inputs at inference	Supplied structure
Integrated, strict	Yes	None	Graph; engine's typed library/policy
Integrated, anchored	Yes	Training labels only	Same; access-matched baseline
Original CFN, no label inputs	Yes	None	Cardinality family and clue membership
Frozen-table reuse	Source only	Declared target anchors	Count family, grid and local compiler
Supplied-law reference	No gradient fitting	Declared anchors	Family, membership and numerical law

5 Constructive consequences: prediction and learned laws

In brief. Near-perfect Minesweeper prediction, natural-task gains beyond ensembling, and preservation across a ten-task panel.

Minesweeper: close the gap and recover the relationship. Under matched strict access, TSGL raises AUC from 97.552 to 99.090; with training-label anchors, from 98.432 to 99.968. These remove 62.8% and 97.9% of the baseline's remaining pairwise ranking error, $1 - \mathrm{AUC}$, computed from the unrounded means. Both protocols fit training targets; only the latter reads training labels as observations. The gains combine a strong preserved predictor with explicit relational computation (Table 5).

The original cardinality-factor network (CFN) makes the learned object readable. Given the count family and clue membership, it fits numerical entries in $\phi_\theta(y_{\mathcal{N}(a)}; c) = T_{\theta,c}(\sum_{j\in\mathcal{N}(a)} y_j)$. It reaches .9906 AUC without label inputs and .9998 with training-label inputs (Table 3). Exercised equality rows peak at the true count; the clipped ≥ 5 row recovers inequality support (Figure 6). The learned relationship is readable; unexercised entries remain unidentified.

Table 3: **Minesweeper performance and declared access.** CFN: ten official splits; TSGL: two confirmation splits. Published GCN* (Luo et al., 2024) and BuNN (Bamberger et al., 2025) provide external context, with reported standard deviations. The small cross-study mean differences are descriptive; the matched TSGL gains are in Table 5.

Model	Information supplied	AUC (0–1)
Published GCN*	Training labels; no label inputs; generic architecture	$.9786 \pm .0024$
Published BuNN	Training labels; no label inputs; bundle diffusion	$.9899 \pm .0016$
TSGL, strict	Training labels; no label inputs; typed library/policy	.9909
CFN	Cardinality family; training labels, no label inputs	.9906
CFN	Cardinality family; training labels also as inputs	.9998

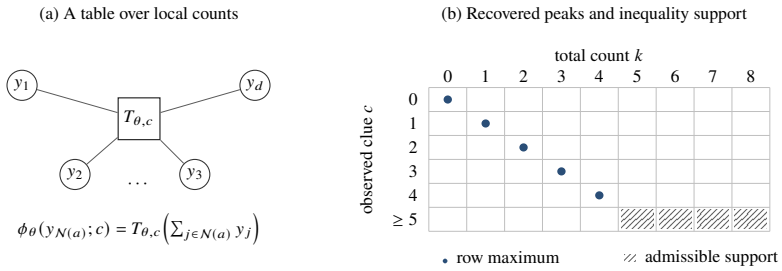


Figure 6: **Read the recovered law.** Equality-clue row maxima match the true count on exercised entries; the clipped clue recovers inequality support. The diagram reports maxima/support, not fitted probability magnitudes.

The benchmark contains substantial deductive structure: sound propagation from the declared clues and training anchors logically entails 97.2% of test nodes. This measures available deduction, not certification of CFN predictions. With the numerical law supplied, exact inference in radius-three local models reaches about .9957 AUC without gradient training. Explicit relationships support both learning and execution.

Preserve the evidence that inference needs. Retaining direction under fixed feature-propagation recipes adds 6.57 accuracy points on Roman-empire and 9.85 on arxiv-year. These contrasts motivate strong directed baselines (Rossi et al., 2024). Appending the raw clue to the same Minesweeper representation adds 6.0 AUC points. The system therefore preserves raw observations alongside neural predictions before testing the value of further computation.

Arxiv-year: structure adds more than another predictor. Arxiv-year uses the OGB citation graph (Hu et al., 2020) with LINKX year targets and official splits (Lim et al., 2021). Across four confirmation splits, TSGL beats both generic correction and a second-GNN ensemble (Table 4). Gains over the ensemble are 1.016 accuracy points strict and 1.559 anchored, positive on three and four splits, respectively; the ensemble wins the remaining strict split by $.083$ points. A descriptive comparison with locked *ungated* ensemble proposals ($64.359/65.212$) leaves gains of $.980/1.363$ points. The advantage survives both added-predictor and acceptance controls.

Table 4: **Arxiv-year: structure improves on correction and ensembling.** Mean accuracy ($\%$) over four official splits of one graph. Every branch uses the later acceptance rule; label access and selection are matched. Per-split values and computational budgets: Appendix B.

Access	Baseline	Generic correction	Two-GNN ensemble	TSGL
Strict	63.556	63.700	64.322	**65.338**
Anchored	64.633	65.734	65.016	**66.575**

The mixture reveals useful complementarity. On the two newer strict splits, the specialist scores 59.658, the baseline 63.672, and their mixture 65.442. Its checkpoint was selected for contribution to the mixture, which outperforms both components. The later comparison reuses saved proposals and previously evaluated acceptance sets under a predeclared control protocol (Appendix B).

Selective computation: nine gains, 31 exact preservations. Across twenty fits and forty dependent evaluations, TSGL improves eight through structured inference and one through generic correction. In the remaining 31 it returns the saved baseline pointwise. All nine accepted deployments improve on test, giving zero observed regressions on this panel (Table 5). On Roman-empire, baseline retention preserves a strong predictor that already uses informative direction. The system evaluates the additional contribution of each candidate relative to that retained predictor.

Table 5: **Validated gains or exact preservation.** Percent native metric; two fits and two access protocols per task. Upper group: nine gains and three Tolokers ties; lower: 28 ties. Arxiv-year uses two splits; Table 4 adds two and applies the later acceptance rule. Sources: Platonov et al. (2023); Yang et al. (2016); Shchur et al. (2018), and arxiv-year above.

Dataset	Metric	Strict: base → TSGL	Anchored: base → TSGL	Deployed path
Validated supplementation in at least one evaluation				
Minesweeper	AUC	97.552 → **99.090**	98.432 → **99.968**	Structured
Arxiv-year	Acc.	63.440 → **65.235**	64.409 → **66.473**	Structured
Tolokers	AUC	84.308 → 84.308	85.117 → **85.409**	Generic (1/4)
Exact baseline preservation in every evaluation				
Roman-empire	Acc.	92.932 → 92.932	93.223 → 93.223	Baseline retained
Cora	Acc.	83.700 → 83.700	84.150 → 84.150	Baseline retained
CiteSeer	Acc.	71.100 → 71.100	71.200 → 71.200	Baseline retained
PubMed	Acc.	79.600 → 79.600	79.600 → 79.600	Baseline retained
Amazon-Computers	Acc.	91.945 → 91.945	91.873 → 91.873	Baseline retained
Amazon-Photo	Acc.	96.144 → 96.144	96.242 → 96.242	Baseline retained
Questions	AUC	77.683 → 77.683	77.667 → 77.667	Baseline retained

Cost and development outcomes. Specialists used about 4.88 of 5.42 hours of confirmation fitting; backbones used 32.5 minutes, excluding development and other overhead. Extensions on filtered squirrel (Rozemberczki et al., 2021; Platonov et al., 2023) and Amazon-ratings (Platonov et al., 2023) stopped before confirmation after missing development gates against generic and ensemble controls (Appendix B).

6 Learn a relationship once; answer multiple queries

In brief. Frozen learned count relationships support joint inference on independent boards without target fitting.

TSGL tests whether relational computation improves a capable predictor. The compiler tests a second consequence: whether a learned relationship remains executable for subsequent questions. Node marginals alone do not determine joint-event probabilities. Direct node-to-joint exports and coherent prototypes did not yield useful learned dependence (Appendix C); the separate count-family compiler provides a positive test with independent target instances and explicit dependence controls.

Within a supplied count family, a source learner alternates an E-step over latent counts with an M-step for an observation table. We freeze the table and reuse it on 16 independent Minesweeper boards, of sizes 100×100 and 160×160, at 50% and 25% anchor budgets. With grid membership and a local compiler supplied, the same object answers three event families: whether a pair contains two mines, whether a neighbourhood contains at least two, and whether it is safe. No target fitting or query-specific learning occurs.

Table 6: **Frozen relationships support new joint queries.** Macro Brier over three event families on independent boards; lower is better. Arms share accepted local support. Untrained-initialization and own-product controls separate learning from dependence.

Predictor	50% anchors	25% anchors
Supplied numerical law	0.019490	0.031530
Learned, identity-initialized	0.019531	0.031670
Learned, random-initialized	0.020949	0.033319
Untrained identity initialization	0.029419	0.046795
Learned model's own product of marginals	0.022196	0.038511

Both learned starts outperform the untrained initialization; the identity-initialized model closely tracks the supplied law. Joint predictions reduce Brier over products of its *own marginals* by 12.0% and 17.8%, with both board-bootstrap difference intervals above zero. This isolates dependence value while holding the individual marginals fixed.

Scores use common supported windows: 1,474 of 1,536 board/center/budget cells (95.96%), with abstention elsewhere. One sparse-anchor query family remains unresolved. From one source-fit regime, the compiler demonstrates local reuse at two sizes: one frozen object answers multiple questions on unseen boards.

7 Related work

Graph rewiring changes propagation connectivity (Gasteiger et al., 2019b); NRI and discrete structure learning infer graph structure for prediction (Kipf et al., 2018; Franceschi et al., 2019). Factor-graph neural networks learn higher-order message computations (Zhang et al., 2020); learned factor graphs fit local models for sum-product inference (Shlezinger et al., 2022); neural-enhanced belief propagation learns corrections to BP messages (Garcia Satorras & Welling, 2021). These supply distinct ways to construct inference. Our contribution is the controlled decomposition: an identical-evidence query intervention, circuit comparisons within a declared GCN budget, and crossed membership/operator/anchor interventions. The exact ranking results explain one obstruction; TSGL and the compiler test predictive supplementation and reusable dependence with explicit controls.

This complements algorithmic alignment (Xu et al., 2020; Veličković & Blundell, 2021). MLPs on fixed aggregation features rival trained GNNs on 12 of 14 benchmarks, with Minesweeper and Roman-empire as exceptions (Rubio-Madrigal & Burkholz, 2026); our interventions test which evidence and computation change their performance. Further comparisons appear in Appendix H.

8 Discussion

Observed relationships and computational dependencies are distinct structures. Our interventions establish the cost of identifying them: query changes alter the required interactions, recompilation improves a common GCN update family, and correct observations can damage a miscompiled circuit. Exact constructions explain why extending reach and creating interactions are different resources.

The constructive consequences are near-perfect Minesweeper prediction, natural-data gains beyond correction and ensembling, and learned relationships reusable across independent instances and queries. TSGL adds inference to strong predictors; the compiler retains executable relationships. Learning membership and building query-specific circuits automatically are the next architectural steps. Selected studies were preregistered; frozen misses and post-hoc controls remain explicit in Appendix F.

The design principle is broader than either implementation: **treat the graph as evidence, and make inference an explicit object of design, learning and evaluation.** An observed edge tells us how the data are related. It does not, by itself, tell us how to answer the question.

USE OF LARGE LANGUAGE MODELS

The authors used large language models to aid and polish the writing; for retrieval and discovery, including finding related work; and to assist research ideation and the execution of experiments. The authors take full responsibility for all content of this paper.

REPRODUCIBILITY STATEMENT

Proofs and their assumptions are in Appendix A. The anonymous code supplement maps the paper's central figures and tables to executable commands, configurations, saved masks and numerical records (Appendix H.2). It includes the controlled interventions, integrated TSGL evaluation and independent-board reuse. Their full protocols are in Appendices E, B and C. Quick CPU checks verify integrity, selected identities and small inference examples; reproducing trained models is a separate operation. Historical appendix results with unavailable original scorers or omitted checkpoints are identified explicitly.

REFERENCES

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *Conference on Learning Theory*, 2023.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural networks for message diffusion on graphs. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.15540.

Maya Bechler-Speicher, Ido Amos, Ran Gilad-Bachrach, and Amir Globerson. Graph neural networks use graphs when they shouldn't. In *International Conference on Machine Learning*, 2024.

Maya Bechler-Speicher, Ben Finkelshtein, Fabrizio Frasca, Luis Müller, Jan Tönshoff, Antoine Siraudin, Viktor Zaverkin, Michael M. Bronstein, Mathias Niepert, Bryan Perozzi, Mikhail Galkin, and Christopher Morris. Position: Graph learning will lose relevance due to poor benchmarks. In *International Conference on Machine Learning*, pp. 81067–81089, 2025.

Zijun Cui, Hanjing Wang, Tian Gao, Kartik Talamadupula, and Qiang Ji. Theory-guided message passing neural network for probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 667–675, 2024.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks*, 2022.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International Conference on Machine Learning*, 2019.

Robert G. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1): 21–28, 1962.

Victor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019a.

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, 2019b.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R. Benson. Combining label propagation and simple models out-performs graph neural networks. In *International Conference on Learning Representations*, 2021.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, 2018.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Renming Liu, Semih Cantürk, Frederik Wenkel, Sarah McGuire, Xinyi Wang, Anna Little, Leslie O'Bray, Michael Perlmutter, Bastian Rieck, Matthew Hirn, Guy Wolf, and Ladislav Rampášek. Taxonomy of benchmarks in graph representation learning. In *Learning on Graphs Conference*, 2022.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic GNNs are strong baselines: Reassessing GNNs for node classification. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Péter Mernyei and Cătălina Cangea. Wiki-CS: A Wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020.

Seth Nabarro, Mark van der Wilk, and Andrew J. Davison. Learning in deep factor graphs with Gaussian belief propagation. In *International Conference on Machine Learning*, 2024.

Anthony Opipari, Chao Chen, Shoutian Wang, Jana Pavlasek, Karthik Desingh, and Odest Chadwicke Jenkins. Differentiable nonparametric belief propagation. *arXiv preprint arXiv:2101.05948*, 2021.

Namyong Park, Ryan A. Rossi, Nesreen Ahmed, and Christos Faloutsos. MetaGL: Evaluation-free selection of graph learning models via meta-learning. In *International Conference on Learning Representations*, 2023.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: are we really making progress? In *International Conference on Learning Representations*, 2023.

Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022.

Tom Richardson and Rüdiger Urbanke. *Modern coding theory*. Cambridge University Press, 2008.

Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M. Bronstein. Edge directionality improves learning on heterophilic graphs. In *Proceedings of the Second Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pp. 25:1–25:27, 2024.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. doi: 10.1093/comnet/cnab014. URL https://academic.oup.com/comnet/article/9/2/cnab014/6271062.

Celia Rubio-Madrigal and Rebekka Burkholz. Fixed aggregation features can rival GNNs. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gSZhNPp103.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018. Relational Representation Learning Workshop, NeurIPS 2018.

Nir Shlezinger, Nariman Farsad, Yonina C. Eldar, and Andrea J. Goldsmith. Learned factor graphs for inference from stationary time sequences. *IEEE Transactions on Signal Processing*, 70:366–380, 2022.

Petar Veličković and Charles Blundell. Neural algorithmic reasoning. *Patterns*, 2(7):100273, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

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

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020.

Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48, 2016.

Zhen Zhang, Fan Wu, and Wee Sun Lee. Factor graph neural networks. In *Advances in Neural Information Processing Systems*, 2020.

A Formal results and their exact scope

A.1 The observation model and affine confinement

Fix a Tanner graph with K observed check signs $\sigma \in \{-1, +1\}^K$. The latent bits are independent Bernoulli(p), $0 < p < 1/2$, and $L_0 = \log((1-p)/p)$. Check-node features contain their signs; bit-node features can contain fixed indicators. More generally, the entire feature array may be any affine function $X(\sigma) = X_0 + \sum_{j=1}^{K} \sigma_j X_j$. Propagation operators are fixed with respect to the observation realization. The readout is linear in any concatenation of propagated features, with an optional bias.

Proof of Theorem 1. For every fixed operator P and integer $r \geq 0$,

$$P^r X(\sigma) = P^r X_0 + \sum_{j=1}^{K} \sigma_j P^r X_j.$$

Taking a node row, concatenating such rows and applying a linear readout preserves this form. The resulting score is $S(\sigma) = \alpha + \beta^\top \sigma$. The argument is unchanged for multiple fixed operators or any training procedure that chooses the readout coefficients. It also allows fixed affine standardization. No graph-size, radius or connectivity assumption is needed. □

The quantifiers matter. Once training has produced a score, the evaluation observation varies while the score's coefficients and preprocessing remain fixed. Normalization recomputed from that same observation can introduce an observation-dependent denominator and falls outside this argument. We do not claim that finite-sample empirical standardization differs by a proved $o(1)$ term in the registered experiments. A theorem for that preprocessing requires additional assumptions and a separate concentration argument.

A.2 When an affine ranking must lose AUC

Let $\mu_1(a)$ and $\mu_0(a)$ be the class-conditional probabilities of atom a, all strictly positive. For independent draws $A_1 \sim \mu_1$ and $A_0 \sim \mu_0$, define

$$\mathrm{AUC}(S) = \Pr\{S(A_1) > S(A_0)\} + \tfrac{1}{2}\Pr\{S(A_1) = S(A_0)\}.$$

A Bayes score ranks atoms by their likelihood ratio $\lambda(a) = \mu_1(a)/\mu_0(a)$.

Lemma 1 (Extreme points of an affine score)**.** *For $S(\sigma) = \alpha + \sum_j \beta_j \sigma_j$, maximizing coordinates satisfy $\sigma_j = \mathrm{sign}(\beta_j)$ whenever $\beta_j \neq 0$. Minimizers are their antipodes. In particular, a unique maximum has its unique minimum at the opposite corner.*

Proof. Each term $\beta_j \sigma_j$ is maximized independently by this choice and minimized by its negative. A zero coefficient leaves that coordinate free, preventing a unique extremum. □

Proof of Theorem 2. For two atoms with $\lambda(a) > \lambda(b)$, ordering a above b rather than below it gains

$$\mu_1(a)\mu_0(b) - \mu_1(b)\mu_0(a) > 0$$

in their paired AUC contribution. Tying them loses half of that gain. Thus achieving Bayes AUC requires respecting every strict likelihood-ratio comparison.

In case (a), an affine score that does not put the unique highest atom u strictly first already loses. If it does, Lemma 1 places its unique minimum at $-u$. But the Bayes minimum is $v \neq -u$, so $\lambda(-u) > \lambda(v)$ while the affine order reverses them. This also loses strictly.

In case (b), the distribution of $S - \alpha$ is symmetric about zero under either class. Therefore $S(A_1) - S(A_0)$ has a symmetric distribution, and its positive and negative probabilities agree. With half credit for ties, AUC is exactly $1/2$. A nonconstant likelihood ratio has a strict improving comparison, so Bayes exceeds $1/2$.

Finally, a finite atom set has finitely many weak orderings. If none of the affine orderings attains Bayes, their best AUC is separated from Bayes by a strictly positive gap. This establishes a gap for the supremum, not merely for each fixed choice of coefficients. □

Table 7: **Exact constructions and reported extensions.** The two higher-arity rows report numerical enumeration at the stated parameter, not a new closed form. At $p = 1/20$, the two-check Bayes AUC is $3971/4000$, its gap is $171/4000$, and the three-check Bayes AUC is $1598147/1600000$. At $\varepsilon = .05$, the parity-gadget Bayes AUC is .905.

Instance	Bayes AUC	Affine comparison
Two-check chain	$(1-p)^2(1+2p)$	Cap $1-p$; strict gap
Three-check chain	$(1-p)^3(1+3p+6p^2)$	Strict gap by criterion (a)
Arity $(3,3)$ chain	Numerically enumerated	Gap at least .0346 at $p = .05$
Arity $(4,4)$ chain	Numerically enumerated	Gap at least .0280 at $p = .05$
Direct two-check star	$(1-p)^2(1+2p)$	Bayes attained; gap zero
Noisy parity gadget	$1-2\varepsilon(1-\varepsilon)$	Every affine score has AUC $1/2$

A.3 An explicit two-check calculation

Let the observations be $\sigma_a = (-1)^{i+j}$ and $\sigma_b = (-1)^{j+k}$ for independent bits i, j, k. In the leaf query, the target is i. The positive score orientation below ranks $i = 0$ above $i = 1$; negating the score gives the equivalent flip-probability orientation.

On this tree, the check-to-variable message from b to j is

$$2\,\mathrm{artanh}(\sigma_b \tanh(L_0/2)) = \sigma_b L_0.$$

The next variable-to-check message is $L_0 + \sigma_b L_0$. Passing it through a multiplies it by σ_a, so the leaf posterior after two sum–product iterations is $L_0(1 + \sigma_a + \sigma_a\sigma_b)$. The product coefficient is nonzero, but the strict-gap result uses the ordering, as well as that algebraic observation.

For two signs, affine orderings are determined by the signs of β_a, β_b, $\beta_a - \beta_b$ and $\beta_a + \beta_b$. The coefficient plane has eight open sectors, eight boundary rays and the origin: 17 weak-order classes. Enumerating the conditional atom masses gives the following seven possible values for $(1-p) - \mathrm{AUC}(S)$.

Slack below $1-p$	Number of weak-order classes
0	3
$1-2p$	3
$(1-2p)/2$	3
$p(1-p)(1-2p)$	2
$p(1-p)(1-2p)/2$	2
$(1-2p)(p^2-p+1)$	2
$(1-2p)(p^2-p+2)/2$	2

Every slack is nonnegative on $0 < p < 1/2$, and three orderings attain zero, including ranking by the target's adjacent check sign alone. Hence the best affine AUC is exactly $1-p$. Enumerating the Bayes order gives $(1-p)^2(1+2p)$, proving Table 1. This finite calculation does not require a trained probe to find the optimal coefficients.

For the center query, the same two checks both touch the target directly. The posterior becomes $L_0(1 + \sigma_a + \sigma_b)$ and is itself affine. Its AUC is the same Bayes value, with no affine gap. This control establishes the stated gadget boundary; it does not make all tasks of a specified graph depth equivalent.

The original verification protocol enumerates atom laws symbolically, compares AUC numerators over positive denominators, counts polynomial roots in the open parameter interval and checks signs at an interior point. It reports 15 of 15 checks passing. The anonymous supplement does not contain that original suite; `checks/audit_numbers.py` supplies separate independent enumeration and arithmetic checks (Appendix H).

A.4 A FINITE-CANDIDATE DISPATCH GUARANTEE FOR THE HISTORICAL RULE

This subsection analyzes the earlier empirical-argmax replacement rule. It is not a guarantee for the stricter paired-SE acceptance rule of the integrated node system in Section 4. Exact fallback identity applies to both, but the near-best-candidate conclusion below requires the algorithm and IID assumptions stated here.

Theorem 3 (Isolated dispatch under independent sampling)**.** *Let $h_0, \dots, h_K$ be predictors fixed independently of m identically distributed, independent dispatch examples. Let $v(h)$ be their population accuracy and $\widehat{v}(h)$ the dispatch accuracy. Deploy an empirical maximizer when it strictly beats h_0; otherwise deploy h_0 exactly. For $0 < \delta < 1$, define*

$$\epsilon = \sqrt{\frac{\log(2(K+1)/\delta)}{2m}}.$$

Then with probability at least $1-\delta$, $v(h_{\text{deploy}}) \geq \max_k v(h_k) - 2\epsilon$. In the refusal branch, predictions are pointwise identical to those of h_0.

Proof. Condition on the candidate set. Hoeffding's inequality and a union bound give $|\widehat{v}(h_k) - v(h_k)| \leq \epsilon$ simultaneously for all $K+1$ candidates. The rule selects an empirical maximizer even when it refuses: refusal means none strictly beats the baseline. Writing k^* for a population maximizer,

$$v(h_{\text{deploy}}) \geq \widehat{v}(h_{\text{deploy}}) - \epsilon \geq \widehat{v}(h_{k^*}) - \epsilon \geq v(h_{k^*}) - 2\epsilon.$$

The identity claim follows directly from returning the saved baseline artifact. □

The observed isolated study compares three saved seed checkpoints, so $K = 3$. An earlier calculation used $K = 1$; correcting it enlarges the reported bound by a factor of approximately 1.08. For Reddit, $m = 11916$, the reported ϵ is $.0146$ and the measured accuracy gap is $.0032$. These are empirical comparisons with the nominal bound. Application of the theorem to dependent graph-node outcomes requires justified independent units or a suitable dependence-aware inequality. Merely splitting node indices does not supply that assumption. AUC is pairwise and requires its own concentration analysis.

A.5 THE BALANCED NOISY-PARITY CONSTRUCTION

Let U, V be independent uniform signs, with target $Y = UV$. Observe $\sigma_1 = UN_1$ and $\sigma_2 = VN_2$, where the independent noise signs satisfy $\Pr(N_i = -1) = \varepsilon$, $0 < \varepsilon < 1/2$. Conditional on either target class, the two observation atoms of each parity are equiprobable, so the class-conditional laws are invariant under simultaneous sign reversal. The observed product matches Y with probability $q = (1-\varepsilon)^2 + \varepsilon^2$. Its binary Bayes ranking has AUC q (including ties); every affine ranking has AUC $1/2$ by Theorem 2(b). The gap is $q - 1/2 = 2(\varepsilon - 1/2)^2$. This declared gadget is not a claim that arbitrary FSM observations have the required symmetry.

A.6 WHAT REMAINS CONJECTURAL

We conjecture that, in the measured law-governed families, low-access single-draw empirical risk minimization in generic message-passing classes tends to remain near the affine probe cap, whereas additional independent realizations can permit approach to a law-aligned reference. This is an optimization-and-access hypothesis, not a consequence of affine confinement. A general proof would need a specified model class, distribution, optimization procedure and asymptotic regime. Connections to parity and leap-complexity analyses are motivating analogies rather than transferred guarantees (Abbe et al., 2023).

B INTEGRATED NODE-SYSTEM PROTOCOLS AND CONTROLS

B.1 DATA ACCESS AND FITTING

The ten-task integrated panel uses the released Minesweeper, Roman-empire, Questions and Tolokers masks (Platonov et al., 2023); public Planetoid masks for Cora, CiteSeer and PubMed; inherited random masks for Amazon-Computers/Photo; and official LINKX arxiv-year splits. The Amazon

masks allocate 60% training, 10% validation, 20% test and leave 10% unused. They must not be compared as if they were 60/20/20. Filtered squirrel in the stopped extension is the deduplicated benchmark, not the original unfiltered graph.

Each inherited validation mask is partitioned equally using seed $99000 + s + 1000j$, where s is the fit seed and j the split ID. Selection and acceptance are disjoint from each other and from training/test. All node features and edges are transductively available; only training labels are optimization targets. Selection labels choose checkpoints and mixture weights. Acceptance labels are read after proposals are saved; test labels score the deployment. These separations are per fit, not claims that historically studied datasets were unseen by the research program.

Table 8: **Actual confirmation fits.** IDs are split/seed. Counts are training/selection/acceptance/test nodes for the first fit in each dataset; masks are saved separately for every fit. Recipe columns show the first selected baseline, not the entire tuning menu.

Dataset	IDs	Counts	Selected recipe
Minesweeper	1/0, 2/0	5000/1250/1250/2500	gcn, 12 layers, 64 hidden
Arxiv-year	1/0, 2/0	84671/21167/21168/42337	dir, 9 layers, 512 hidden
Tolokers	0/0, 0/1	5879/1469/1470/2940	gcn, 9 layers, 256 hidden
Roman-empire	1/0, 2/0	11331/2832/2833/5666	dir, 5 layers, 512 hidden
Cora	0/0, 0/1	140/250/250/1000	appnp, 1 layers, 512 hidden
CiteSeer	0/2, 0/3	120/250/250/1000	appnp, 1 layers, 512 hidden
PubMed	0/0, 0/1	60/250/250/1000	appnp, 1 layers, 512 hidden
Amazon-Computers	1/0, 2/0	8251/687/688/2750	sage, 4 layers, 64 hidden
Amazon-Photo	0/0, 0/1	4590/382/383/1530	gcn, 6 layers, 256 hidden
Questions	1/0, 2/0	24460/6115/6115/12231	sage, 6 layers, 512 hidden

The baseline menu reuses recorded GCN/GraphSAGE recipes with their learning rates, normalization, residual connections and feature preprocessing. Planetoid tasks additionally allow row-normalized features and APPNP (hidden 512, dropout .8, learning rate .01, weight decay .0005, budget 2000). Roman uses directed SAGE depth 5; official arxiv uses depth 9, width 512, batch normalization, residual connections, dropout .5, learning rate .001 and budget 2500. All winners are selected without acceptance labels. Released configuration records retain the complete per-fit menu and selected epoch; this is not a shared hyperparameter setting across every dataset.

Specialist inputs concatenate $X, \hat{A}X, \hat{A}^2X$. Directed graphs additionally provide incoming/outgoing one- and two-step features, including mixed compositions. This same raw/propagated feature bank enters generic heads and typed candidates. Generic heads have two hidden layers, width 128 or 256 and dropout .3. Typed candidates have width 128, three update families and ten rounds with damping .5. For $C \leq 10$, a factor takes at most eight distinct neighbours per role, selected by decreasing raw-feature cosine similarity after canonical edge deduplication; ties have deterministic ordering. A single affinity policy was selected over canonical membership on development CiteSeer, Minesweeper and arxiv-year cells. It is frozen for confirmation and invariant to edge storage order. For $C > 10$, four damped uncapped pairwise compatibility updates replace the higher-order engine. Thus the class-count branch changes the approximation, not just memory use.

Specialists use AdamW, learning rate .003, weight decay .0005, at most 600 updates, gradient clipping 1, selection checks every 10 steps and stopping after 200 steps without improvement once 200 steps have run. A decaying type-entropy term has initial coefficient .01 and vanishes at step 300. Every checkpoint is scored through its best mixture weight in $\{.25, .5, .75, 1\}$; a reported standalone specialist is therefore *not* checkpoint-selected for standalone accuracy. In anchored fitting, half the training nodes are randomly exposed as observations per update and the rest are prediction targets. At evaluation all training labels may be inputs. The label-input perceiver is warm-started from the strict baseline and adapted for 300 steps. Correct&Smooth uses 50 correction and 50 smoothing steps and selection-chosen $\alpha \in \{.5, .8, .95\}$. The best of strict, label-input and C&S predictions becomes the preserved anchored baseline.

B.2 Local tables and message updates

For a center label ℓ and neighbour class channel c, the encoder emits local log-compatibility tables $\theta_{v,\ell,c}(k)$. The cardinality table indexes the count k; the affine restriction is $a_{v,\ell,c} + b_{v,\ell,c}k/D$; parity indexes $k \bmod 2$. These tables depend on the center's raw/propagated features through the shared encoder. A learned task-level softmax mixes the three types in log-message space. It is a type mixture, not the perceiver–specialist mixture of Equation (3).

Cavity beliefs subtract the previous factor contribution. For a neighbour slot j, the count kernel computes the leave-one-out count distribution $Q^{-j}(k)$ of the other binary class-channel indicators. After averaging exponentiated compatibility over the center's cavity label belief, write the resulting table as $G_c(k)$. The message is

$$m_{c,j} = \log \sum_k Q^{-j}(k) G_c(k+1) - \log \sum_k Q^{-j}(k) G_c(k).$$

The parity kernel instead flips the parity index in the first contraction. Prefix/suffix contractions avoid enumerating all assignments at each update. The implementation also computes the center's factor contribution, sums incoming messages into unary logits, and uses the declared damping and numerical clipping. Incoming and outgoing roles have separate table heads. Per-class binary-channel contractions and loops make the complete multiclass recurrence an approximation; exact local contractions do not establish a globally exact or coherent joint predictor. Anchors clamp permitted training-label logits during the recurrence. The alternative $C > 10$ branch uses four iterations of a learned incoming/outgoing pairwise compatibility correction to its raw unary logits.

B.3 Selection and acceptance

The system selects one best generic or structured proposal on selection data. It separately records the best generic control. For paired accuracy differences d_i, $\widehat{\mathrm{SE}} = \mathrm{sd}(d_i)/\sqrt{m}$. For binary AUC, form each positive example's ranking placement against negatives and each negative example's placement against positives. For proposal-minus-baseline placement differences d_i^+, d_j^-, use

$$\widehat{\mathrm{SE}}^2 = \frac{\mathrm{var}(d^+)}{m_+} + \frac{\mathrm{var}(d^-)}{m_-}.$$

Accept if the native-metric gain is at least .002 and greater than $c\widehat{\mathrm{SE}}$, with $c = 2.24$ for the ten-task panel. This formula does not account for arbitrary graph dependence; no coverage certification is asserted. Refusal returns the original saved array, verified pointwise. Replay of saved execution paths agrees to within 9×10^{-7} on the checked paths.

Table 9: **Component performance and contribution to the system.** Native metric in percent, two fits per dataset. Checkpoints were selected for mixture performance. These rows report each specialist separately from the deployed predictor; the complementarity comparison in the main text evaluates its contribution to the mixture.

Dataset	Metric	Strict specialist	Anchored specialist
Minesweeper	AUC	99.023	99.968
Arxiv-year	Acc.	60.928	65.040
Tolokers	AUC	81.696	80.831
Roman-empire	Acc.	46.294	73.641
Cora	Acc.	77.000	73.150
CiteSeer	Acc.	60.050	61.350
PubMed	Acc.	70.700	64.500
Amazon-Computers	Acc.	90.873	76.345
Amazon-Photo	Acc.	78.562	84.706
Questions	AUC	68.147	65.417

B.4 THE LATER ARXIV ENSEMBLE COMPARISON

Reuse the saved baseline and specialist proposals on official confirmation splits 1–4. Train a second GNN from an independent initialization under the selected baseline recipe and the same masks; construct its anchored counterpart using the same label-input/C&S choices. Select its mixture weight on selection data. The later comparison uses $c = 2.50$ for every reported branch, including the earlier saved generic/typed proposals. It does not change the original panel's scores or frozen outcomes. Each new strict GNN has 9,533,445 parameters; additional fitting across its strict/anchored paths takes about 12–18 minutes per split. The comparison matches access, not parameters or total compute.

Table 10: **All paired arxiv confirmation results.** Accuracy in percent. Repeated splits and the two access protocols share one graph, so these eight rows are not independent graph populations.

Split	Access	Baseline	Generic	Ensemble	TSGL
1	Strict	63.887	63.887	64.194	65.659
2	Strict	62.992	63.247	64.894	64.811
3	Strict	63.965	64.284	63.965	65.864
4	Strict	63.380	63.380	64.235	65.019
1	Anchored	64.256	65.489	64.740	66.337
2	Anchored	64.563	65.784	65.612	66.608
3	Anchored	64.872	66.060	64.872	67.015
4	Anchored	64.839	65.605	64.839	66.341

Typed minus ensemble has split SD .8644 points strict and .4694 anchored. We report all paired values rather than infer a population interval from dependent split/access cells. The ungated comparison in the main text is a post-hoc diagnostic on already-locked proposals, with no refitting or reselection. Prior directed-engine results around 67.8% used different candidate fitting and selection; they are not substituted into this table.

B.5 DIRECTION, REPRESENTATION AND STOPPED BREADTH

The earlier anchored direction-typed engine, separate from integrated TSGL, gives $50.1523 \rightarrow 67.7842$ accuracy on five official arxiv-year splits when the factors separate citing and cited roles. Its paired mean gain is 17.6319 points with split-bootstrap interval $[17.4627, 17.7873]$. This official-split reproduction is descriptive; the seven-of-seven frozen prediction score belongs to the earlier random-split study. Both anchored arms fit training targets and use training-label observations. The official no-label-input direction-typed arm scores 65.0202; these means belong to this earlier fitting/selection protocol, not Table 4.

Fixed-perceiver direction contrasts use ten official Roman-empire splits, five official LINKX arxiv-year splits, and ten filtered squirrel/chameleon splits (Rozemberczki et al., 2021; Platonov et al., 2023). Roman gain is 6.57 points (paired-split bootstrap interval [6.28, 6.84]), arxiv 9.85 [9.69, 10.01], squirrel 5.04 [3.02, 7.19], chameleon .50 [-2.61, 3.34]. These intervals describe variation across splits of the same graph.

Wikipedia record linkage and label-free verification. The original recovery script matched filtered rows to geom-gcn raw rows using exact binary feature vectors and dataset labels. All dataset labels entered these record keys, including validation/test nodes; they were not model inputs. A post-hoc reconstruction removes labels from both matching-key expressions and resolves duplicate feature vectors injectively by neighbourhood agreement with the filtered graph. It reproduces the saved raw-index mappings, directed edge arrays and split masks exactly for both datasets. Feature vectors alone uniquely identify 764 of 890 chameleon rows and 2,046 of 2,223 squirrel rows; the remaining candidates are resolved by observed graph structure. The resulting directed graphs have 13,516 and 65,577 edges, respectively.

Before repair, symmetrizing retained hyperlinks recovers all 46,998 squirrel edges and 8,853 of 8,854 chameleon edges; the single remaining chameleon edge is represented in both directions. The final undirected edge sets agree exactly. This check verifies connectivity consistency; equality to the saved directed arrays additionally shows that labels were unnecessary to reconstruct the inputs actually

evaluated. It is a post-hoc provenance check, not a new trained-model experiment or a revision of the frozen scores. The anonymous supplement includes the recovery implementation and its feature-only verification entry point.

On squirrel, a post-hoc same-recipe symmetrized Dir-SAGE control gives 36.6 versus 38.0 with direction, +1.37 [.73, 1.99]. The tuned undirected incumbent versus a Roman-recipe directed model instead changes both recipe and direction and is not an isolating contrast. Chameleon has no resolved same-recipe direction effect. Direction-aware graph learning is established prior work (Rossi et al., 2024); these experiments measure its importance within this evidence/circuit decomposition.

The broad representation census was a component study, not the complete integrated system. A finite-budget replacement engine on tuned representations often loses. A separate exploratory fold-basis comparison evaluates the engine against the predictor that supplies its representation: on 15 of 18 tasks, the native-metric change is within one point. On Roman-empire's directed fold basis, the earlier engine adds .12 points, with a paired-split interval $[.01, .24]$. These are measured increments from this engine, not bounds on all possible remaining headroom. The current directed baseline already retains the informative relation. The raw-clue bypass on Minesweeper gives 99.9 versus 93.9 AUC with the same representation retained. A linear reconstruction R^2 is only a linear-readout statistic; these experiments do not prove that the representation contains no recoverable evidence. Benchmark design and protocol choices deserve separate attention (Liu et al., 2022; Bechler-Speicher et al., 2025).

The later filtered-squirrel and Amazon-ratings extensions used split 0 and seeds 10/11 for selection-only development. Continuation required typed mixtures to beat baseline, generic and ensemble controls in both seeds, with mean margin at least the larger of .003 and paired-seed spread. Neither task passed; acceptance and test values were not read and confirmation was not launched. No looser post-hoc rule rescues them. Recorded ten-task confirmation fitting is 19,507.9 seconds in total, with about 17,556.1 seconds on specialists and 1,951.8 on backbones; development, data preparation, repeated inference and other overhead are excluded.

Published reference provenance. The separate published tuned GCN reference is $97.86 \pm .24$ AUC; Luo et al.'s final Table 3 reports GraphSAGE* $97.77 \pm .62$ (Luo et al., 2024). BuNN reports $98.99 \pm .16$, a ten-seed mean and standard deviation (Bamberger et al., 2025). Platonov et al. report GCN 89.75 and SAGE 93.51 and tune aggregation depth over 1–5 with their other stated hyperparameters fixed (Platonov et al., 2023). Published, project-rerun and access-matched integrated scores are distinct.

Remaining-ranking-error reduction. For the integrated Minesweeper comparison, the relative reduction is $(\mathrm{AUC}_{\mathrm{TSGL}} - \mathrm{AUC}_0)/(1 - \mathrm{AUC}_0)$, using mean AUCs. Strict means .975515 and .9909025 give 62.8446%; anchored means .984315 and .999678 give 97.9471%. Ranking error here is $1 - \mathrm{AUC}$, counting ties with half weight in positive/negative pair orderings. It is not node-classification error, and the ratio of mean errors is not a mean of per-split relative reductions.

C Independent-board reuse and the boundary on joint integration

C.1 What is learned and frozen

Within the supplied clipped-count family, the observation model is a 6×6 table $T_{s,c} = \Pr(c \mid s)$ for latent clipped count s and observed clue c. Inference uses Bernoulli cell priors, anchor observations and local count chains, followed by exact variable elimination subject to a width cap. The source learner uses radius-two models for its E-step, six EM iterations and additive smoothing .1 in the M-step. The identity-biased initialization is $.5I + .5/6$; the supplied-law comparator is I. Random initialization uses five Dirichlet row starts on 1000 source clue locations and selects by source evidence likelihood. The cell prior is estimated from permitted source training labels. These are source labels, not target-board fitting.

The independent-board study freezes one preselected source split's learned tables and prior. It does not quantify source-fit uncertainty. Board seeds 981000–981015 generate 16 independent boards, eight at each size 100 or 160. Mine probability is .2, clue visibility .5, counts clip at 5, and nested independent uniform masks define the .5/.25 anchor budgets. Forty-eight centers per board are

selected without labels, away from boundaries, with the pair-query members unanchored at both budgets. The target labels enter scoring only; the inference document receives labels only where the mask permits observations.

Each board/center/budget cell asks for a pair event, a neighbourhood count-at-least-two event and a zero-neighbour-mine event. Own-product controls use marginals from the same inferred distribution; the count control is a Poisson-binomial calculation. All arms try radius 3 and, on a width refusal, radius 2. If any arm refuses at that radius, all retry on the smaller common model. If neither succeeds, the cell is recorded as refused. There is no prediction or accuracy claim outside the common accepted support. Of 1536 cells, 1474 are accepted: 914 at radius 3 and 560 at radius 2. This denominator is not a count of independently sampled nodes.

C.2 Uncertainty and controls

The independent unit is the generated board. Resampling boards retains all paired arms and events within each board. At 50% anchors, identity-initialized learned minus supplied-law macro Brier is .000041 with 95% interval [-.000055,.000133]. The random-initialized model is worse than the supplied law by .001459 [.000637,.002333]; useful learning does not make all initializations identical. Both learned starts outperform the *untrained identity-biased* comparator. This is not a comparison against each random start's own untrained predictor.

Own-product minus learned-joint macro Brier is .002665 [.002185,.003133] at 50% anchors and .006840 [.005770,.007875] at 25%. The sparse-anchor pair family alone is unresolved. The size change in excess-over-supplied error is -.000002 [-.000207,.000192]. These are local reuse results, not proof of generalization to longer inference chains, unknown laws or arbitrary graphs. Recorded wall time is 114.97 seconds with 24 CPU workers; the reused source job's 1694.31 seconds includes its earlier evaluation and is not a pure training cost.

An earlier study reused tables across masks on the same board. Source training masks included 1213–1274 of each target mask's 2500 test labels; those cross-mask arms do not establish independent-instance transfer. Their same-split comparisons remain valid under declared access, but they are not pooled with this study. The independent generated boards here remove that source/target overlap.

C.3 What the node-to-joint attempts did not establish

The first exporter interpreted node-trained parameters as a local energy and normalized it exactly. On Minesweeper, base-product Brier was .042116, the local model's product .125674, and its joint .120661. The exported probabilities differed from the successful node predictor by about .318 in mean absolute probability difference, not Brier or accuracy. On Questions, local joint deployment was .085105 versus .076147 for the neural query control. The local interpretation did not reproduce the node model's useful behavior; discriminatively trained potentials can define a joint distribution, but that fact alone does not establish a good one.

A later globally coherent forest mixture preserved its own marginals without earning useful structured deployment. The bounded Tolokers local model then used one stable sensor during fitting and execution and IPF to preserve its marginals. Of two seeds, one accepted a .25 mixture. Its query Brier was .125205, compared with .129181 for the full-perceiver product. A *post-hoc* no-refit control at the locked weight, removing learned interactions, scored .125107; the product of the deployed model's own marginals scored .125071. The other seed refused and returned the baseline exactly. That deployment gain did not require learned dependence. Normalization and successful acceptance are not substitutes for this control.

The successful count-family compiler and these failed integrations have different training, context and inference procedures. They do not form a matched causal test of changing the objective alone. The general node/joint unification problem remains open.

D PROTOCOLS, ACCESS DECLARATIONS AND ESTIMATOR HISTORY

D.1 RESTRICTED DIAGNOSTIC AND ITS LIMITS

Write $R(r)$ for the native-metric score of a fitted linear head on fixed propagated features at radius r. The historical rules use endpoint gain $\Delta_R = R(r_{\text{last}}) - R(1)$ and reference headroom $H_{\text{ref}} = R_{\text{ref}} - R(r_{\text{last}})$. Where no computable reference exists and $\Delta_R < .05$, the rule labels saturation. Where $H_{\text{ref}} \geq .05$ and $\Delta_R < .02$, it labels plateau; otherwise slope. These native-metric thresholds are not universal constants. A low radial gain does not establish closeness to the best possible learner, and an unavailable reference is undefined, not zero.

A registered four-configuration GNN menu supplies a finite comparator, not a ceiling. Circuit-cone and epidemic-time statistics are query-matched diagnostics, but not necessarily affine functions of the same sign encoding. Only the specified linear members satisfy Theorem 1. Correct&Smooth stays within the registered envelope plus spread on eight of nine tested tasks; Actor is the exception. Fixed GDC improves the tested plateau probe by only.002. Neither observation is an impossibility result for learned rewiring or nonlinear graph models.

Effective realizations (ER) count declared local-law supervision: labeled bits times independent draws for LDPC, or observed adjacent anchor pairs for the FSM. They are family-specific counts, not mutual information. A measured FSM sweep orders together under matched counts, while a proposed universal per-table-entry normalization fails (Appendix E). Classical decoding and trellis references follow their stated observation models (Gallager, 1962; Richardson & Urbanke, 2008; Rabiner, 1989). Where exactness is not established, they remain principled references rather than Bayes optima.

D.2 NAME THE PROBE THAT PRODUCED THE RESULT

Version	Optimization / metric	Role in the record
RProbe-v1	LBFGS, at most 200 steps; binary AUC	Original 20-cell functional-validation panel
RProbe-v2	Adam, 300 steps; binary AUC	Prospective panel on 18 tasks
LRGB extension	Adam, 300 steps; macro-F1	PascalVOC-SP (Dwivedi et al., 2022); radii 0 through 8
RProbe-public	Adam, 300 steps; native metric	Named package interface with provenance object

The ordinary radius set is $\{0, 1, 2, 3, 5\}$; factor-circuit studies additionally reach radius 99. The node and factor probes use symmetric normalization of $A + I$. In v2, weight decay is validation-selected from $\{10^{-4}, 10^{-2}\}$. Circuit cells use a directed-forward-cone statistic; epidemic cells use radius-restricted infection-time statistics. Those estimators are related diagnostics but are not all members of the affine class of Theorem 1.

Finite fitted curves can decrease with radius. The alternative estimator $\widehat{R}^*(r) = \max_{s \leq r} R(s)$ changes exactly one of eighteen frozen regime labels: squirrel, whose original $\Delta_R = .0499$, moves to slope. We retain the original scoring rather than retrospectively treating the monotone variant as the preregistered instrument.

Legacy LDPC weight-decay selection used the evaluation split. A correction uses a 40/20/40 train/validation/test split and validation-only selection. The $(4, 8)$, $n = 1200$ cell gives $\Delta_R = .0004$ and reference headroom .086; the $(3, 6)$, $n = 4800$ cell gives $\Delta_R = -.0013$ and headroom .059. Both retain the plateau label. The corrected labels do not make the original selection procedure valid.

D.3 FUNCTIONAL VALIDATION: RETAIN THE PROVENANCE LIMIT

The earlier instrument-validation gate concerns the area *above* the curve, not its endpoint gain. On a 20-cell panel of 12 circuit cells and 8 datasets, the recorded rank correlation against internal depth-structure values is .908 for the area functional and .086 for the endpoint. The original executable scorer did not survive.

The deterministic replacement audit uses a mechanically sourced target vector from registered run files. Nineteen cells are usable; one dataset lacks a registered target. Its Spearman correlations are .82 for area and .26 for endpoint. The leave-one-circuit-out range is .80–.82; circuits alone give .62; datasets alone give $-.71$ on targets with very little variance. This supports mixed-panel ranking, not a common within-family effect. Neither the area functional nor the original .908 defines the frozen regime labels; those use radial gain and reference headroom.

D.4 The generic envelope and interaction probes

The registered generic menu contains four configurations: GCN with three layers; GraphSAGE with three layers; a family-specific deeper/wider GraphSAGE (four layers and hidden width 512, or nine layers); and APPNP (Kipf & Welling, 2017; Hamilton et al., 2017; Gasteiger et al., 2019a). Each is screened for 300 steps. The validation winner runs three seeds at the full budget with best-validation checkpointing and patience. Envelope spread denotes the reported maximum-minus-minimum at convergence; it is not a confidence interval or a universal architecture envelope.

The nonlinear probe is a 64-unit ReLU hidden layer on the same F_r as the linear probe. It uses Adam at learning rate .01 for 300 steps, validation-selected weight decay from $\{10^{-4}, 10^{-2}\}$, per-radius standardization and seed 0. The quadratic probe uses a fixed seed-0 random projection to 64 dimensions, appends all 2080 pairwise products and fits a logistic head under the same optimization protocol. Neither trains a graph representation. Post-hoc replications use seeds 0–9; seed 0 reproduces three of four examined frozen task values exactly, and the frozen WikiCS value lies inside the replicated seed range.

D.5 Transformer configurations and the fitting check

The registered GPS-class arm (Rampášek et al., 2022) uses features plus eight Laplacian positional coordinates; three GPSConv layers of hidden width 64; a GraphSAGE local block; two-head global attention; dropout .3; and a linear head. Optimization is Adam at .001, weight decay 10^{-5}, validation checkpointing and patience, with three seeds. Four further variants change one architectural choice: six layers; width 128 with four heads; sixteen positional coordinates; or removal of the local block (attention-only).

Thus the detailed protocol specifies *five total configurations*, including attention-only. The stronger fitting diagnostic doubles capacity and tests learning rates .003 and .01 for 3000 steps. Both variants reach training-anchor accuracy 1.0000, but test accuracies fall to .0978 and .0502. This rules out an explanation based solely on inability to fit those anchors; it does not prove that every optimization schedule must fail.

D.6 Task families and observations

LDPC decoding Regular (d_v, d_c) Gallager ensembles use a configuration model without duplicate edges. Independent bit flips have probability $p = .05$. The observations are syndromes and the target is bit-flip ranking, scored by AUC. Syndrome-domain sum–product runs 50 iterations. It is exact on the tree gadgets and a principled reference on loopy ensemble graphs. The access count is labeled training bits multiplied by independent draws.

Finite-state machines A random deterministic transition table δ has C states and S symbols. A length-n walk emits observed symbols corrupted with probability $\varepsilon = .05$. Anchors reveal states at density ρ. The ordinary target is non-anchor state recovery. A trellis forward–backward solver supplied with the true δ and ε provides the exact reference. The clean query intervention additionally reveals true symbols at the same anchors to both query arms. Its access count is observed adjacent anchor pairs.

Circuit faults ISCAS'85 netlists are subjected to single stuck-at faults, with observation density .05. Signature matching identifies a fault exactly only when its observed signature is unique. Both selective accuracy and all-detected-fault metrics must be reported with their denominators.

Epidemic sources The observation process is discrete susceptible–infected (SI), not SIR, with sparse observation times on real topologies. The target is source finding, scored by MRR. Dynamic message passing is a principled reference whose performance can itself be limited by the observation process.

Node benchmarks The development suite spans 15 standard benchmarks, including Cora, WikiCS, Actor, WebKB tasks and Reddit (approximately 233,000 nodes and 41 classes), loaded through PyTorch Geometric (Mernyei & Cangea, 2020; Pei et al., 2020; Hamilton et al., 2017; Fey & Lenssen, 2019). The prospective containment subset has eleven node tasks. These counts describe different panels, not competing totals for one panel.

D.7 What each arm is told

Label	Supplied information	Typical arm
L0	No law family or law parameters	Generic end-to-end network
L1	Typed library or declared factor family; infer permitted types/parameters	Learned typed engine / CFN
L2	Factor scaffold; infer model parameters	Compiled-circuit GCN
L3	Governing law and parameters	Trellis, BP, DMP, signature matching

These labels expose information differences; they do not turn the comparison into an equal-information architecture contest. They are access declarations rather than a formally nested hierarchy. In particular, Minesweeper's supervised no-label-input CFN receives the cardinality family. Supplying a family and learning its parameters is different from discovering an arbitrary factor family without prior structure.

E Additional measurements and controlled comparisons

E.1 Reproducing the circuit intervention

The reference implementation is `src/h26/compile.py`, with the residual GCN and scorer in `src/h11/strong.py`. The saved record is `results/h26/compile.json`; these paths are relative to the anonymous supplement (Appendix H.2). They identify the circuit experiment, not the later integrated TSGL fits.

Data and supplied scaffold. Use PyTorch Geometric's `HeterophilousGraphDataset`, Minesweeper, with its official 50/25/25 masks for splits 0–4 and model seed 0. The input has 10,000 variable nodes, seven-dimensional one-hot features, and 39,402 undirected edges after symmetrizing, deduplicating and removing self-loops. Feature zero marks an unrevealed clue; the other six channels encode counts $0, 1, 2, 3, 4, \geq 5$. Exactly 5,000 clues are revealed. All arms fit the same training targets, and no target labels are input features or anchors.

For each revealed node v, create factor node f_v connected in both directions to $N_E(v)$, excluding v itself. The compiled graph contains only these variable–factor edges, not the original variable–variable edges: 15,000 nodes and 39,404 undirected incidences. Variables retain their original seven features. A factor receives a copy of its host's six observed count channels, $X[v, 1{:}7]$. Compilation uses the supplied interpretation that a clue describes its observed neighbours, with the host excluded. Membership and factor features are derived from adjacency and the revealed clue; no hidden count or target label is consulted. "Fixed evidence" therefore means unchanged observed values and masks, with an explicit deterministic lifting into a circuit whose semantics are supplied.

Network and fitting recipe. For every graph, propagation is the row-normalized $P = D^{-1}(A + I)$. Hidden updates share

$$h^{\ell+1} = h^{\ell} + \mathrm{Drop}_{.2}\big(\mathrm{ReLU}\big(\mathrm{LN}\big(P\,\mathrm{Linear}_{\ell}(h^{\ell})\big)\big)\big)\,.$$

The evidence graph uses $h^0 = \mathrm{ReLU}(WX + b)$. The bipartite version has one input linear map per node type and a learned type embedding, followed by ReLU; a common linear head predicts two

logits. Factor nodes are masked out of training and evaluation. Widths match the 135,426-parameter reference to within 0.7% (Table 11). These are a shared update family and matched parameter budget, not identical encoders or depths.

Training uses full-batch cross-entropy, AdamW at learning rate $.003$, weight decay $.0001$, at most 1,500 updates per fit, and validation AUC every ten updates. Stop after more than 300 updates without improvement and restore the best-validation state. The compiled depths 8 and 16 are preregistered; one depth is chosen by mean best-validation AUC across the five splits: $.942690$ versus $.953784$, selecting 16. The evidence arm has depth 8. The menu's stated rationale is that variable–factor–variable communication traverses two edges. This counts communication paths, not identical nonlinear updates or spatial receptive fields: factors can connect co-neighbours. Both depths are therefore reported explicitly. The update cap is per fit, not an equal total search cost across the two-depth and one-depth menus.

Table 11: **Exact architecture and result accounting for the circuit intervention.** Means over official splits 0–4, seed 0. The eight-layer contrast isolates the depth-matched comparison; the sixteen-layer compiled arm is the validation-selected headline.

Circuit	Layers	Width	Parameters	Mean test AUC
Evidence adjacency	8	128	135,426	.903524
Compiled membership	8	127	134,495	.943693
Compiled membership	16	90	135,632	.955917
Random factor members	16	90	135,632	.505651
Clique completion	8	128	135,426	.742937
Degree-preserving rewire	8	128	135,426	.508880

Randomization and scoring. For random membership, independently sample each factor's original number of distinct variable members uniformly without replacement, using NumPy seed 2000. This preserves factor counts, factor degrees and total edge count, but not variable degrees or the complete topology. Retain the true factor features and use the selected compiled depth. Clique completion adds all missing pairs within each revealed host's neighbourhood to the original graph, producing 95,230 undirected edges. Rewiring uses NumPy seed 1000 and rejects self/duplicate edges while making 394,020 accepted double-edge swaps; the variable degree sequence is checked exactly. All five splits share each seed-defined control graph. The historical scorer assigns sequential ranks to sorted positive softmax probabilities, without averaging tied ranks; the archived scorer is included for literal reproduction. No experimental scores are recomputed by the manuscript build.

Cora control and deduction strata. In the earlier Cora control, replacement changes accuracy by -3.3 percentage points and augmentation by $+.25$. Its code uses seed-defined 60/20/20 masks, factor features $\log(1 + \deg v)$, and keeps the original edges only in the augmentation arm. Absolute Cora scores and uncertainty are not substituted from the later integrated panel.

Nodes can also be stratified by the round at which a specified sound constraint-propagation procedure determines them. Reported GCN accuracy falls approximately $.89 \to .78 \to .70$ across rounds 1–3, with similar accuracy across the tested 2-, 4-, 8- and 16-layer models. The available summary does not supply four independent curves or justify statistical equivalence, so we do not draw them. With anchors clamped, the aligned engine remains at 1.0 through round 4+. These rounds are not proved minimum depths of all algorithms; anchor availability, clue density and other stratum differences can also vary. They distinguish the stated procedure's composition requirements from receptive-field reach without identifying every cause of the stratified gap.

E.2 All factorial cells

The $2 \times 2 \times 2$ experiment changes membership, local operator and anchors. Table 12 retains the complete means behind Figure 4; the two panels use their own native metrics and are not pooled into one effect size.

Table 12: **Factor membership, operator and anchor interactions.** Minesweeper means are over five official splits; FSM means are over five draws. "Clamped" enforces observed anchors, and "none" removes them. Membership randomization follows the family-specific constructions below.

		Minesweeper AUC		FSM accuracy	
Membership	Operator	Clamped	None	Clamped	None
True	Typed	.9997	.9907	.7792	.7414
True	Affine	.8903	.8872	.1443	.1233
Random	Typed	.4992	.7033	.1473	.1483
Random	Affine	.5156	.5355	.1424	.1233

E.3 The clean query intervention and its pilot

Generator, graph and shared evidence. The entry point is `src/h48/e8_query.py`; `src/h39/fsm.py` supplies the generator and path edges, and `src/h31/perceiver.py` supplies normalization. Each draw has $n = 20{,}000$ positions, C states and $S = 4$ symbols. Draw every transition-table entry $\delta(s, x)$ uniformly from the C states, draw s_0 and symbols x_t uniformly, then set $s_{t+1} = \delta(s_t, x_t)$. Corrupt each symbol independently with probability .05 by uniformly replacing it with a different symbol. Select anchors independently with probability .02. Each non-anchor position independently enters validation or test with equal probability. Generator and trace RNGs both use the draw's declared seed: 4242 for D0; 9101, 9102, 9103 for D1–D3; 9201 for D4. D0–D3 use $C = 8$ and D4 uses $C = 16$.

The probe graph is an undirected path on time positions, with edges $\{t, t+1\}$ and added self-loops during normalization. The single float32 input tensor has row

$$X_t = [\mathrm{onehot}(\widetilde{x}_t);\ a_t\,\mathrm{onehot}(s_t);\ a_t\,\mathrm{onehot}(x_t);\ a_t],$$

where a_t is the common anchor indicator. Its width is $2S + C + 1$, namely 17 or 25. Hidden state and true-symbol channels are zero away from anchors. The implementation constructs X once and passes the same tensor object, same path and same masks to both target fits. Supervision is s_t versus x_t at those same anchor positions. Neither probe receives the numerical transition table or the noise parameter.

Probe and aggregation. Use $P = D^{-1/2}(A + I)D^{-1/2}$ and concatenate $[X, PX, \ldots, P^r X]$ for $r \in \{0, 1, 2, 3, 5\}$. Standardize each concatenated feature across all positions with a standard-deviation floor of 10^{-6}. At each radius, fit one linear softmax head on anchors for 300 Adam updates, learning rate .01, seed 0; choose weight decay from $\{10^{-4}, 10^{-2}\}$ by final validation accuracy. The reported probe value is the maximum test accuracy across the registered radii, a descriptive curve envelope rather than a validation-selected deployed radius. All radii remain in the result JSON. The optional generic comparison uses a three-layer, width-256 residual GCN with BatchNorm, dropout .5, Adam .01, zero weight decay, gradient clipping at 1, at most 800 updates, validation checks every ten updates, patience greater than 200 updates, and seeds 0/1.

Reference access. A single forward–backward calculation is supplied the true δ, corruption probability .05, uniform initial-state and symbol priors, and the same observations. Revealed symbols clamp their emission likelihood rows; revealed states clamp the forward/backward states. State and symbol posterior modes are read from that one clamped trellis. Thus the reference knows a numerical law that the probe must infer from its permitted training targets. It is exact for this finite-state observation model, including the final symbol's observation-only likelihood. Table 13 reports the empirical fitted-probe/reference comparisons, not a theorem of finite-sample probe optimality.

Table 13 contains the published four-decimal values underlying Figure 2. Each draw uses one evidence array and identical split masks for both tasks. The reported flip is the state-reference gap minus the symbol-reference gap. Rounding can make the difference computed from displayed entries disagree in the final digit with a separately rounded reported flip.

The earlier pilot fixes the generative draw and anchor locations but reveals query-specific anchor labels. Its state probe reaches .2151, the trellis .779 and the generic GCN .250; its symbol probe

Table 13: **Byte-identical evidence, changed target.** D1–D3 are unseen draws and D4 is an unseen configuration. P19–P21 require a flip of at least .15 on D0 and at least .10 on the unseen draws and configuration. All three predictions pass. The five symbol gaps of at most .001 are an additional, unscored observation.

		State query		Symbol query		Reported
Draw	(C, S)	Probe	Reference	Probe	Reference	flip
D0	$(8, 4)$	.2017	.7827	.9512	.9515	$+.581$
D1	$(8, 4)$	.1742	.7048	.9489	.9494	$+.530$
D2	$(8, 4)$	.2090	.8173	.9512	.9515	$+.608$
D3	$(8, 4)$	.1445	.7375	.9504	.9514	$+.592$
D4	$(16, 4)$	.1177	.6541	.9490	.9494	$+.536$

and ceiling both reach .9512. Since the input values differ, that pilot is not the controlled identical-evidence intervention. The clean experiment supplies both sets of revealed anchor values to both tasks.

E.4 The factorial: contrasts and implementation controls

All sixteen cell means appear in Table 12. The Minesweeper factorial uses five official splits, distinct from the ten-split final system sweep. The FSM factorial uses five draws with a fixed loopy-BP solver and validation-selected affine parameter θ per cell, without gradient training of a generic graph model.

Minesweeper implementation. `h48/factorial.py` uses the `h22/lcfn.py` count-factor implementation: width 128, neighbourhood cap 8, ten inference rounds and damping .5. Random membership replaces the neighbour-list matrix by its seed-0 row permutation, while keeping the host features fixed. Typed tables are unrestricted over center label and neighbour count; the affine arm restricts the count dependence to an affine function. Fit each cell with seed 0, AdamW at .003, weight decay 10^{-5}, binary cross-entropy, gradient clipping 1, at most 600 updates, validation every ten, and patience greater than 200 updates. Clamped training reveals a random half of training labels and predicts the remainder, with logit clamps ± 12; validation/test execution reveals all training labels. The no-anchor arm fits all training targets and never inputs them.

FSM implementation. `h48/e9_factorial.py` uses the $(8, 4)$ table generated by seed 4242 and trace seeds 0–4 at $n = 20{,}000$, anchor probability .02, and noise .05. True factors connect $(t, t + 1)$; randomized factors connect $(t, \rho(t))$ for a seed-0 permutation ρ, dropping self-pairs. This does not preserve the chain topology. Typed potentials use the supplied transition table and channel likelihood at the source position; affine potentials are $\theta \mathbf{1}[s_u = s_v]$ for validation-selected $\theta \in \{.25, .5, 1, 2\}$. All cells use uniform message initialization, 60 log-domain BP iterations and damping .5. Anchors clamp state unaries where allowed. No gradient fitting occurs; affine-parameter selection remains a validation operation.

Under clamped anchors, the membership effect (true minus random) is .5005 with typed Minesweeper factors and .3747 with affine factors. For FSM it is .6319 with typed factors and .0019 with affine factors. Under true membership, clamping adds .0090 to the typed Minesweeper arm and .0378 to the typed FSM arm. Under randomized membership, clamping changes the typed arm by $-.2041$ and $-.0010$, respectively. These contrasts separate an operator advantage from its interaction with membership and observations.

The FSM true-membership, typed, clamped cell reaches .7792, close to the independently implemented true-trellis deployment value .7789 in the wrong-law study. This is a cross-implementation accuracy check; it does not establish that sixty BP rounds recover every full-chain marginal exactly.

E.5 Interaction-order probes and post-hoc seed studies

Table 14: **Changing the readout without learning a graph representation.** All heads use the same propagated feature family. A closure fraction above one is possible because the envelope is an observed model menu, not an optimum. Cornell's reported closure fraction is 1.50.

Task	Linear	MLP-64	Envelope	Interpretation at this budget
Cornell	.5946	.7027	.6667	Gap closes; quadratic probe only .4865
Texas	.7027	.8378	.8468	Reported closure fraction .94
Wisconsin	.7451	.7647	.7451	Closes within reported spread

Across eleven held-out node tasks, the linear probe's best value lies within the envelope spread on six; on Actor it exceeds the envelope by 2.3 points. The MLP closes only .090 of the remaining squirrel gap and .108 on BlogCatalog. Ten-seed BlogCatalog results remain 2.5 points below the envelope band throughout.

Two post-hoc studies explain, but do not rescore, frozen misses. WikiCS has frozen nonlinear gain .0121, above a .01 bar; the ten-seed mean is .0098 with 95% interval $[.0076, .0120]$, and nine of ten seeds fall inside the envelope band. Coauthor Physics gives $.9722 \pm .0004$ against a menu band of $.9714 \pm .0003$; nine of ten seeds exceed the upper bar .9717. The latter is a reproducible exceedance of the finite menu, not an impossibility for that menu's architectures under every training procedure.

On PascalVOC-SP (Dwivedi et al., 2022), the linear probe goes from $R(0) = .0446$ to $R(8) = .0455$ in macro-F1, with $R(8) - R(1) = .0006$. The MLP reaches .0581 and the quadratic probe .0627. The MLP improvement over the linear best is .0126, missing the frozen .05 prediction. These measurements identify unsuccessful probe families, not a proof of irreducible function-class difficulty.

E.6 Transformer breadth, reported arm by arm

Tables 15 and 16 distinguish the cross-system comparison from the four-variant sweep. The training-anchor row makes the two failure modes visible without inferring them from test accuracy alone.

Table 15: **Breadth-study comparison.** Entries are accuracy. The WikiCS generic envelope has reported spread $\pm.0075$. "Best GT" is wide on the FSM cells and deep on WikiCS. Missing entries are not inferred.

Cell	Chance	Probe	Generic	Best GT	Typed	Exact
FSM transductive, ER $= 5$	.1251	.2027	.2551	.1965	.6308	.8149
FSM inductive, ER $= 245$	.1250	—	.2891	.1822	.7804	.7831
WikiCS	—	.7852	.7985	.7555	—	—

Table 16: **Four variants beyond the registered GPS arm.** Each variant uses three seeds. The registered GPS arm reaches .7527 on WikiCS. Only two of four variants show the predicted perfect-anchor-fit/poor-test signature, so the prediction of at least three is a miss.

Measurement	Deep	Wide	PE-16	Attention-only
FSM test, ER $= 5$	.1706	.1965	.1648	.1510
FSM test, ER $= 245$	.1551	.1822	.1645	.1424
Training anchors, ER $= 245$	.7253	.7942	1.0000	1.0000
WikiCS test	.7555	.7424	.7395	.6841

E.7 The access-normalization failure

Table 17: **No universal per-entry knee in the registered window.** The study uses fifteen rungs and two seeds per rung. The knee is the first reported access level at which performance reaches at least .9 of the exact reference. All lie below the preregistered window $[.5, 4]$; the smallest family is at ceiling from the first rung. Pooled Spearman .326 fails the .6 criterion.

(C, S)	D	Trace-length range	ER range	Knee ER$/D$	Max spread
$(4, 2)$	8	600–12,000	1–25	$\leq .125$	.0191
$(8, 4)$	32	2,500–50,000	4–130.5	$\simeq .39$	.0468
$(16, 4)$	64	5,000–100,000	12.5–243.5	$\simeq .38$	.0083

The earlier positive observation is narrower: measured FSM transductive and inductive cells order together by ER (Spearman 1.0), and the matched-access LDPC difference is .0088. A count that aligns a limited sweep need not identify all the supervision a solver can extract. Nonadjacent anchors constrain a small transition table even when the adjacent-pair count is one. The failed normalization is therefore not restored by renaming the count.

E.8 The observation-limited branch

Epidemic studies force a third distinction alongside representation and learning access: the observation process can limit even a principled inference procedure. The decomposition study scores two of three predictions correctly and reports dynamic message passing restored to twice the probe floor under the tested observation change. A proposed three-feature ER collapse fails, including its power follow-up; a capacity test scores zero of two. Fixed rules on informative statistics outperform learned rules at low supervision in those experiments. The experimental record supports interacting limitations, not a numerical minimum of quantities on a common scale. We retain the bottleneck account as an organizing principle only.

F Frozen predictions, strict scores and the negative record

F.1 The 54-prediction prospective panel

The list contains 20 tasks. LastFMAsia and its epidemic variant were unavailable at source before any probe ran, leaving 18 evaluated tasks. Prediction dependence is represented by 19 scorer-defined task clusters; those keys need not coincide one-to-one with the displayed task names.

Family	Frozen	Correct	Frozen content and observed limitation
Fault localization	12	12	Unique-signature accuracy 1.0 on four held-out circuits; typed at least generic, with a twofold result on c 6288. Exactness is coverage-qualified.
Saturation containment	33	32	Eleven tasks, three related predictions each. Accepted gain at most .5 points; final soft gate below .15. Observed gate at most .045; one spread-band violation remains a miss.
Access escape	2	2	Ensemble-access escape on new $(4, 8)$ codes and $(3, 6)$ at $n = 4800$.
Plateau immobility	4	2	Affine obstruction survives. Generic learned-arm behavior is not universally immobile, particularly at high supervision.
Epidemic source finding	2	1	A miss separates observation/reference limits from learner-access limits.
Size extrapolation	1	0	Refuted as stated.
Total	54	49	Accuracy .907; Brier .1139.

Equal weighting of the six families yields accuracy .662 and Brier .203. A 10^4-draw task-cluster bootstrap gives interval $[.804, .982]$ for prediction-weighted accuracy. Leaving out containment gives the lowest leave-one-family-out accuracy, .810. Fault localization and containment together score

$44/45$; the four extrapolation families score $5/9$. There is no held-out node task predicted to be nonsaturated in this panel, a coverage gap rather than a successful test of that regime.

F.2 Three preregistrations and a strict correction

Preregistration A covers P 1–P 13: nonlinear probes, isolated dispatch, transformer breadth, the query pilot and the LDPC correction. B covers P 14–P 18: the Minesweeper factorial and wrong-law refusal. Its membership axis was amended from a degenerate identity condition to random membership before any run; that amendment belongs in the immutable record. C covers P 19–P 25: the clean shared-evidence query intervention and second-family factorial.

The shipped scorer for P 17 accepted any worst typed condition with randomized membership. The frozen statement named the *unanchored* random-membership typed condition. The observed worst cell was anchored. Literal scoring therefore changes P 17 from pass to miss. The strict standing score for P 1–P 18 is $12/18$, Brier $.2106$; the preserved lenient result is $13/18$, Brier $.1828$. P 19–P 25 score $6/7$, Brier $.1471$. Together, the strengthening study has eighteen passes and seven misses, listed below. Post-hoc explanations do not change these scores.

Table 18: **Every strict miss in the strengthening study.** The right column describes later interpretation, not a revised verdict.

ID	Frozen bar and outcome	Follow-up and retained implication
P 2	WikiCS nonlinear gain at most $.01$; observed $.0121$.	Ten-seed mean $.0098$, interval $[.0076, .0120]$, nine of ten seeds within the envelope band. The frozen single-seed forecast still misses.
P 3	PascalVOC-SP nonlinear gain at least $.05$; observed $.0126$.	No rescoring or resolving follow-up. The inexpensive radius and readout probes do not close this task.
P 4	Probe no higher than Coauthor Physics envelope plus spread $.9717$; observed $.9722$.	Ten-seed mean $.9722 \pm .0004$, with nine of ten above the bar. This finite model envelope is not a ceiling.
P 5	At least ten of eleven dispatch decisions persist under isolation; observed eight.	Withdrawn acceptances are Actor, squirrel and Reddit; all had zero original test gain. The correction exposes validation reuse, but does not satisfy the forecast.
P 9	At least three of four new transformers show perfect anchor fit and poor test accuracy; observed two.	Two variants memorize; two also fail to fit. A subsequent fitting check does not retroactively make the single-mode prediction correct.
P 17	Worst typed cell predicted to be random membership without anchors; anchored random membership is worse.	Strict miss replaces a lenient pass. A newly frozen directional claim in FSM (P 25) succeeds: $.1473 \leq .1483$.
P 24	FSM anchor effect must reach the specified twofold bar $.0382$; observed $.0378$.	Miss by $.0004$. The larger membership/operator interaction does not depend on passing this separate threshold.

F.3 Later bounded studies: keep outcomes separate

Study	Correct/scored	Brier	What the score does not hide
Independent-board count reuse	6/6	.060	One source-fit regime and common accepted support
Integrated ten-task system	5/6	.2204	Strict Minesweeper .990903 misses the frozen .992 bar
Arxiv replication and perception checks	4/6	.1021	Perception upgrade and Computers gate miss; one conditional non-run
Independent-GNN control and transfer gates	3/4	.1344	Both breadth gates fail; two conditional non-runs
Local node-to-joint export	0/4	.2344	No successful integrated joint reuse
Coherent forest prototype	2/4	.0963	Preservation/boundary hits, not useful dependence
Stable local-joint prototype	3/5	.1350	Correctness, preservation and one acceptance hit; both useful-query predicates miss

Conditional non-runs are not counted as wins or losses; failed prerequisites remain scored. The studies share data, mechanisms and development history and are not pooled into one program-wide success rate. In particular, the historical 111-forecast aggregate has unresolved per-metric inclusion masks and is omitted. This record supports procedural traceability, not a claim of well-calibrated probabilities.

The earlier representation census has literal scores 1/11 and 4/13, with a failure falsifier fired; its fold-basis reinterpretation is post-hoc. The Roman direction study scores 5/8 literally, not its post-hoc 7/8 interpretation; one frozen item had no probability. The arxiv direction study scores 7/7 on its registered random splits, with official-split replication descriptive. The event-query study scores 1/6; same-board learned-law reuse scores 5/7 and does not prove independent-instance transfer. The filtered direction study scores 3/6, including one vacuous hit caused by a negative reference gain; its matched-recipe correction is exploratory. These records are retained rather than silently counting useful measured effects as successful forecasts.

F.4 Tests that could overturn the empirical account

The registered challenges are: a newly preregistered L 0 arm, at the declared low-ER budget in a measured family, closing at least half the reference headroom; a probe-tight task ($\Delta_R < .05$, envelope gap below .01) on which a new architecture beats the menu by more than twice its registered spread; a matched-ER access-mode effect exceeding .05 of exact-reference performance; more than 30% regime mislabeling in a held-out family under the frozen protocol; or a new factorial in which typed operators benefit randomized membership as much as true membership.

These challenges concern the empirical regularities, not the proved affine theorem. Conversely, exceeding a genuinely exact *population* optimum on its declared information and identifiability domain would signal inconsistent assumptions, implementation or evaluation, rather than greater-than-Bayes inference. A finite test estimate above a reference is not, by itself, proof of leakage.

G Original CFN, table recovery and historical dispatch

G.1 Cardinality factors and logical entailment

The original CFN learns tables indexed by an observed clue and the number of mines among its factor members. It is given the cardinality family and true clue membership, not the completed numerical table. The ten official split means are reported in Table 19. These are different fits and a different branch from the integrated results in Table 5.

The BuNN reference is the published ten-seed mean and standard deviation. The CFN no-label-input mean exceeds it by .0007 AUC, or .07 percentage points; this numerical comparison is not a paired significance test. CFN additionally receives the count family and clue membership. These references situate its benchmark performance; the controlled recompilation experiment and integrated panel separately identify circuit value and matched system gains.

Table 19: **Original CFN branch, not the integrated panel.** Minesweeper AUC. Published references are separate from the matched circuit intervention and the project's own fitted arms. Historical dispersion terms for CFN are not assigned an unsupported confidence-interval interpretation.

Model	Supplied information / supervision	AUC
Unary	No relational factor	.5115
Affine pairwise	Pairwise factor class	$.8934 \pm .0059$
Published tuned GCN*	Training labels; no label inputs (Luo et al., 2024)	$.9786 \pm .0024$
Published tuned GraphSAGE*	Training labels; no label inputs (Luo et al., 2024)	$.9777 \pm .0062$
Published BuNN	Training labels; no label inputs (Bamberger et al., 2025)	$.9899 \pm .0016$
CFN	Count family; training labels, no label inputs	$\mathbf{.9906} \pm .0008$
CFN	Count family; training labels also as inputs	$\mathbf{.9998} \pm .0001$

Under the original benchmark's observed clues and revealed training labels, sound constraint propagation determines a reported 97.2% of test nodes, with zero observed contradictions among the determined nodes. Without training anchors the propagation study determines 57.0% of all nodes. These denominators differ: the 97.2% figure is neither label-free coverage nor a certificate for the CFN's predictions. It is not an AUC ceiling. Later posterior-thresholded coverage at probabilities below .01 or above .99 is a separate confidence statistic, not logical entailment.

For equality clues $c = 0, \ldots, 4$, covering 98.7% of clued nodes, every exercised table row has its maximum at the true count. The clipped ≥ 5 bucket recovers the corresponding admissible support. Unexercised entries remain unidentifiable. Figure 6 summarizes reported maxima/support rather than reconstructed probabilities.

A different follow-up on Questions and Amazon-ratings finds stable signs in extracted factor rules. On Questions, a two-parameter rule retains at least half the engine gain on nine of ten splits; Amazon's extracted rule is standalone-predictive against its null. These small-effect findings are not Minesweeper count-table identification and are not used to enlarge its claim.

G.2 Known-law local inference

Supplying the numerical observation law removes the table-learning problem. Exact variable elimination in radius-three and radius-four local models gives reported mean AUC about .9957 and .9982, without gradient fitting. The local window and width-limited fallback are part of the model: exact local marginals are not guaranteed to equal whole-board marginals. The integrated node campaign also records a supplied-count loopy-BP arm; its local contractions are exact but its global inference is approximate. Neither supplied-law branch earns credit for a learned-only system comparison.

G.3 The earlier isolated replacement rule

The historical dispatch study, separate from the current mixture/threshold procedure, compares three saved candidate checkpoints with a saved perceiver on a held-out validation half. Ties refuse. Ten of eleven node tasks refuse, and Coauthor CS accepts with zero measured test gain. Eight of eleven decisions persist after isolation, not the predicted ten. Actor, squirrel and Reddit lose prior acceptances that had yielded zero test gain. These observations do not justify reusing validation for both checkpoint selection and final acceptance.

Table 20: **Utility, not law purity.** Historical FSM deployment study. A wrong table is refused; an imperfect noise model can still improve on the baseline.

Candidate	Validation	Deployed test	Decision
Perceiver	.261	.255	Baseline
True transition law	.7833	.7789	Accept
Wrong transition table	.2318	.255	Exact fallback
Wrong noise model	Not reported	.6561	Accept

The earlier soft-gated system's weight measurements and aggregate gates are not a theorem about the current deployment rule. The IID accuracy lemma in Appendix A applies only to the argmax procedure it explicitly states.

H Reproducibility and reporting conventions

H.1 Positioning the two constructive demonstrations

Table 21 compares the objects that methods learn and execute, rather than assigning broad capability checkmarks. The last two rows are distinct implementations. Having both in a research program does not show that one system unifies them.

Table 21: **Different learned objects support different computations.** Selected comparisons, not an exhaustive capability audit. No row implies that related work cannot be extended to other queries.

Method	Supplied structure	Learned or selected object	Execution / evaluation
GDC (Gasteiger et al., 2019b)	Graph and diffusion construction	Downstream predictor	Prediction on transformed connectivity
Neural relational inference (Kipf et al., 2018)	Interaction-model family	Latent edges and dynamics	Neural message-passing dynamics
Discrete structure learning (Franceschi et al., 2019)	Graph-learning hypothesis class	Structure and GNN parameters	Bilevel learning for prediction
MetaGL (Park et al., 2023)	Candidate model set and graph descriptors	Model selection rule	Select a graph learner
FGNN (Zhang et al., 2020)	Factor-graph representation	Neural variable/factor message updates	Higher-order prediction and approximate inference
Learned factor graphs (Shlezinger et al., 2022)	Stationary Markov factorization	Local conditional models	Reuse in sum-product and Viterbi inference across sequence lengths
Neural-enhanced BP (Garcia Satorras & Welling, 2021)	Factor graph and BP computation	Neural message corrections	Hybrid inference, including channel decoding
Integrated TSGL here	Graph, factor library, bounded membership policy	Perceiver, local tables, mixture proposal	Node prediction with held-out acceptance
Count-family compiler here	Count family, grid membership, local inference language	Source observation table	Frozen-table reuse for three event families

These methods, together with differentiable and Gaussian belief-propagation approaches (Opipari et al., 2021; Cui et al., 2024; Nabarro et al., 2024), establish learned factors and hybrid inference as useful ingredients. Here the experiments test a specific decomposition of graph-learning tasks and its consequences. The compiler's contribution is evidence of frozen-table transfer and dependence value against its own-marginal controls in a declared count family; it is not the first use of a learned local model inside probabilistic inference.

H.2 Anonymous code and reviewer entry points

The anonymous `anonymous_code.zip` supplement is organized around this paper, not the larger research program. Its `README.md` maps figures and tables to source, configurations, commands and reported numerical records. `EXPERIMENTS.md` gives complete commands and dependencies; `DATA.md` gives public dataset sources and exact split conventions. Run `python reproduce.py list` to browse entry points and `python reproduce.py verify` for the no-training checks.

Paper result	Entry point after `python reproduce.py`
GCN circuit intervention	`run circuit --phase mine`
Shared-evidence FSM queries	`run query`
Membership/operator factorials	`run factorial-mines`; `run factorial-fsm`
Integrated ten-task TSGL panel	`panel-all` (use `--dry-run` to list fits)
Arxiv ensemble controls	`run arxiv-controls`; requires the listed proposal fits
Independent-board reuse	`run reuse confirm`; `run reuse score`

Copied implementations and their local dependencies are under `src/`; experiment IDs are retained to preserve imports. `configs/` identifies the integrated fits; `inputs/` includes exact saved masks, official arxiv-year split indices and the small frozen learned count tables. `results/` holds the reported numerical records, separate from new outputs. Supporting entries cover original CFN learning, direction controls, known-law inference and the unsuccessful joint extensions. Neural checkpoints and full datasets are not bundled; the main rerun paths fit their own models using the public data. The reuse path executes the shipped frozen tables, with optional source refitting documented separately.

Only copies were edited for portability and anonymity. Machine paths and identity-linked comments were removed; frozen design documents carry visible provenance redactions. Numerical designs, predictions, thresholds and outcomes are unchanged. `SOURCE_PROVENANCE.json` retains original and shipped hashes; `MANIFEST.sha256` identifies the shipped files. Hash guards verify the anonymous copies, not an unmodified historical file.

H.3 What the source package verifies

The quick verifier checks file integrity, Python syntax, shipped preregistration hashes, selected result arithmetic and lightweight CPU imports/inference. `python checks/audit_numbers.py` independently enumerates the small ranking constructions and checks selected displayed values. The copied raw results are distinguished from the paper's transcribed table summaries. Neither verification path reruns the trained-model experiments or certifies every scientific claim.

Full reproduction requires the listed public datasets, dependencies and, for the neural fits, a CUDA GPU. Checkpoint-consuming controls run after their stated source fits. The historical .908 diagnostic scorer remains unavailable; its replacement audit is a different calculation. Some historical readout/dispatch code requires original checkpoint collections omitted from this compact release and is marked as inspection-only. These exceptions do not remove the central intervention, TSGL and independent-board rerun paths. Environment and hardware differences can change floating-point reductions and checkpoint selection; the original records are not promises of bitwise repeatability.

H.4 Reproduction order and access

For the integrated node system, preserve the recorded dataset version, edge treatment and inherited masks. Recreate the selection/acceptance partition before fitting. Fit the perceiver and the generic/structured candidates with the declared training-label access; lock checkpoints, candidate and mixture weight before acceptance. Return the saved baseline on refusal, then score test outputs. Report baseline, candidate, selected mixture and deployed predictor separately.

For independent-board reuse, freeze the source table and prior before generating target boards. Reconstruct the supplied count family and grid membership, record anchor masks, and compile all compared arms on common supported local windows. Save refusals as well as answers; evaluate event queries only on the declared common support. Exact inference within a finite model is distinct from exact inference about the whole board.

For the historical diagnostic panel, freeze its probe version, thresholds, forecasts and candidate menu before testing the candidate models. The integrated node campaign does not retrospectively inherit confirmatory status from those earlier forecasts.

H.5 METRICS, UNCERTAINTY AND PROVENANCE

AUC, accuracy, macro-F 1 and MRR retain their native meanings. Percentage points are 100 times a difference on the unit scale. Split means on one graph are not independent populations; seed variation is not a substitute for independent-domain replication. The independent-board bootstrap uses boards as units. Other paired-split intervals describe the tested split variability only.

The original generic-menu spread is its reported maximum-minus-minimum; some historical $\pm$ terms lack a resolved SD/SE/CI definition and are not relabeled here. New tables state their aggregation explicitly. Neither an exact fallback nor a high-confidence prediction supplies an unconditional guarantee about accepted specialists.

Only the named frozen studies are prospective. Development choices, corrected scorers, post-hoc seed analyses, the locked ungated ensemble comparison and the no-law mixture diagnostic retain their respective statuses. The unresolved program-wide forecast aggregation is excluded rather than presented as a calibration credential.