WHEN 99% IS FAILURE: DIAGNOSING THE APPROXIMATION–FEASIBILITY GAP IN NEURAL CONSTRAINT SOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural solvers for constraint problems can satisfy nearly all constraints of an instance yet never return a feasible assignment. We exhibit this approximation–feasibility gap in a controlled setting using random 3-SAT, where clause density produces cohorts with different coordination demands. At 3.7 clauses per variable—an intermediate, constrained cohort in our density sweep—we evaluate four closely related GNN configurations, each across three independent training runs. All satisfy 97–100% of clauses on average, yet three never return a feasible assignment while OPTGNN-CLAUSE succeeds on about half of the formulas. We ask why approximation-equivalent architectures differ in exact success, and how that separation changes across the density sweep. Our analysis crosses three axes: architecture, the concept linearly accessible from a network—here *support*, how firmly each variable's current value is held in place by the clauses it alone satisfies—and generator-defined difficulty regimes. A crossed ablation shows that constraint-level aggregation and the relaxed-label output/loss configuration yield feasible solutions only in combination; aggregation alone makes support accessible in a form that transfers across formulas even in a configuration that solves nothing, and this accessibility persists at densities where solving collapses. NAE-SAT reproduces the pattern, coloring preserves the behavioral advantage on a planted family, and Max-Cut bounds its scope. Evaluating neural solvers therefore requires varying architecture and regime jointly and reporting feasibility alongside approximation, with representation analysis explaining what a design adds and what it still lacks.

1 INTRODUCTION

Neural solvers for discrete constraint and partition problems often optimize a continuous relaxation with a graph neural network (GNN) and decode its final states into a discrete assignment (Tönshoff et al., 2021; Yau et al., 2024). This pipeline invites two notions of success that can diverge dramatically. *Approximation* asks what fraction of the constraints an assignment satisfies; *feasibility* asks whether it satisfies all of them. An assignment satisfying 99% of a formula with thousands of clauses still violates dozens, and a solver that reliably reaches this level may never return a valid assignment. We call this discrepancy the *approximation–feasibility gap*. We study it in 3-SAT, NAE-SAT, 3-coloring, and Max-Cut, using 3-SAT as the running example.

The gap is sharp even within one closely related solver family. On random 3-SAT formulas with $n = 1000$ variables and $3.7n$ clauses—the intermediate, constrained cohort of our density sweep—we evaluate four GNN configurations, each across three independent training runs. All satisfy between .970 and 1.000 of clauses on average, yet three never return a feasible assignment while OPTGNN-CLAUSE succeeds on about half of the formulas. This raises a question that approximation alone cannot answer: why do approximation-equivalent architectures differ completely in exact solving, and why does that answer change with instance difficulty?

Neural combinatorial optimization typically compares architectures by objective value, solution quality, and scale (Kool et al., 2019; Khalil et al., 2017; Li et al., 2018; Tönshoff et al., 2021; Yau et al., 2024). Representation studies instead ask what task-relevant information a network encodes (Shoham et al., 2024; Zhang et al., 2025), while average-case CSP theory explains how combinatorial structure

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.

and algorithmic difficulty shift with the parameters that generate instances (Mézard & Zecchina, 2002; Achlioptas & Ricci-Tersenghi, 2006; Achlioptas et al., 2005). The opening question requires these views together: the solver's *architecture*, the problem-relevant *concept* organized in its representations, and the *instance regime* on which that organization is tested.

We vary architecture through a cumulative hierarchy built from the relaxation-based OPTGNN (Yau et al., 2024). Successive configurations add recurrent refinement, an output trained to penalize relaxed constraint violations, and explicit communication from constraints back to their variables. OPTGNN-CLAUSE is the configuration combining the latter two components. To understand what this communication organizes, we examine information that plausibly helps a solver repair its remaining violations. In 3-SAT, a variable's *support* is the number of clauses in which its literal is the only true one (Flaxman, 2008; Vilenchik, 2007). Flipping a highly supported variable breaks several clauses unless other variables move with it, so support measures the local stability of the current assignment.

We ask whether this stability is organized consistently in the network's representations. Holding a trained model fixed, we fit a linear readout of support from variable states on one collection of formulas and apply it unchanged to new formulas. Successful transfer means that the same direction recovers support across instances. Whether such information enables a complete solution can depend on the instances. We define a *regime* as a cohort sampled from a fixed generator at specified parameters, such as random 3-SAT at a given clause density. Although these problems are NP-hard in the worst case (Cook, 1971; Karp, 1972), sparse, denser, and near-critical random cohorts have different solution-space structure and challenge different algorithms (Mézard & Zecchina, 2002; Achlioptas & Ricci-Tersenghi, 2006; Achlioptas et al., 2005). We call these easy, intermediate, and critical regimes; the 3.7-density formulas above lie in the intermediate regime. NAE-SAT and 3-coloring have analogous density-dependent structures (Achlioptas et al., 2001; Bapst et al., 2014; Alon & Kahale, 1997).

Crossing the three dimensions resolves the paradox (Figure 1). On easy instances nearly every configuration solves. At intermediate difficulty the configurations remain matched in approximation, yet only OPTGNN-CLAUSE regularly returns feasible assignments. The crossed ablation identifies a configuration-level interaction: constraint aggregation with the standard objective and the relaxed-label output/loss configuration without aggregation each solve essentially nothing, while their combination solves regularly. Aggregation nonetheless makes support transferable even with the standard objective, separating the representation result from exact success. In the critical regime, support remains recoverable from OPTGNN-CLAUSE's states after its solve rate falls to zero. Thus accessibility separates architectures in the intermediate regime, while within OPTGNN-CLAUSE it does not track success across regimes.

The account extends beyond 3-SAT with problem-dependent strength. NAE-SAT reproduces the architecture-level pattern, and its local-stability signal likewise persists as solving declines with density. 3-coloring preserves OPTGNN-CLAUSE's behavioral advantage on a planted family. Max-Cut provides a boundary case: all assignments are feasible, so we measure attainment of a target cut value, and the connection between local stability and success loosens. Across the four problems, high objective quality can therefore conceal failure to produce the required discrete solution or attain the desired target.

The practical consequence is a three-axis standard for evaluating neural combinatorial solvers. Unless approximation is itself the endpoint, average objective value should be paired with exact feasibility or target attainment, architectures should be compared across generator-defined regimes, and representation analysis should test the concepts that distinguish their behavior. This makes representation analysis useful for explaining architectural progress and identifying what a new solver still lacks.

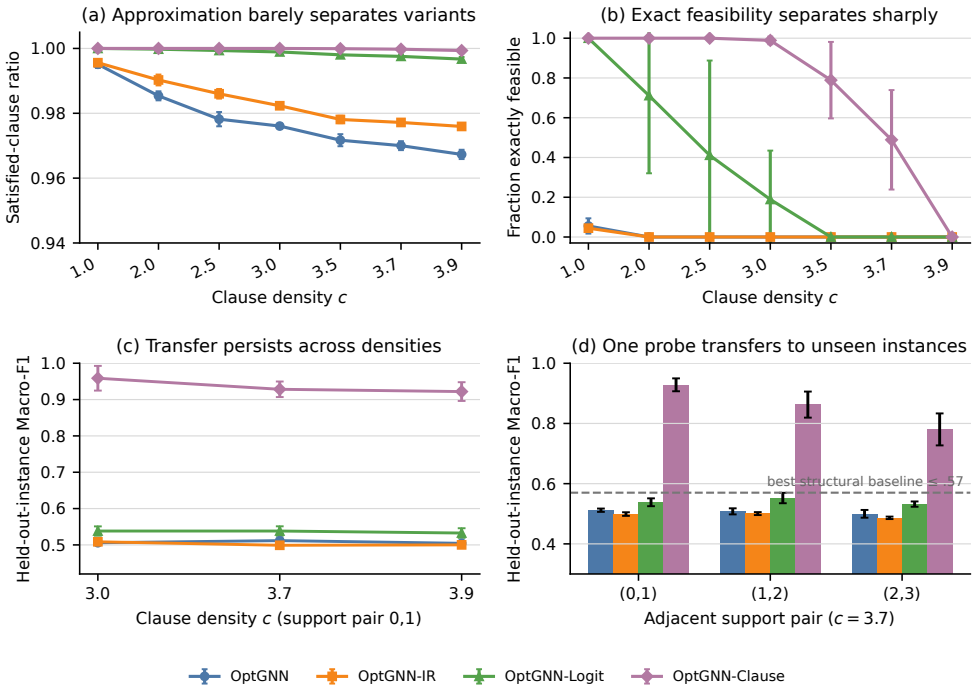


Figure 1: The approximation–feasibility gap and SAT-support accessibility across random 3-SAT cohorts ($n = 1000$; $c = m/n$). (a–b) Mean satisfied-clause ratios remain .967–1.00, while exact feasibility separates sharply; curves use three independently trained models and bars show between-model variation. The separate four-model feasibility evaluation used with the probes is reported in Section 4. (c–d) A linear probe is trained on 20 formulas and applied unchanged to 20 disjoint formulas for four independently trained models. Panel (c) tracks support 0-vs-1 accessibility across density; panel (d) tests all adjacent pairs at $c = 3.7$. OptGNN-Clause exhibits reproducible above-control support accessibility; the dashed line is the strongest structural baseline.

2 Background and Related Work

Related work. Message-passing and relaxation-based neural solvers combine variable and constraint information to produce discrete assignments (Gilmer, 2017; Zhou et al., 2020; Selsam et al., 2019; Lemos et al., 2019; Tönshoff et al., 2021; Yau et al., 2024). Their continuous objectives can obscure exact feasibility: for example, a random assignment already satisfies $7/8$ of 3-SAT clauses, while repairing the remainder can require coordinated changes (Håstad, 2001; Achlioptas & Coja-Oghlan, 2008). Separately, average-case CSP analyses organize instances by generator-defined regimes (Mézard & Zecchina, 2002; Achlioptas & Ricci-Tersenghi, 2006; Achlioptas et al., 2005), classical SAT algorithms exploit local stability signals such as support and backbones (Nudelman et al., 2004; Achlioptas & Coja-Oghlan, 2008; Lincoln & Yedidia, 2020), and neural-solver analyses probe hidden states for task-derived statistics (Shoham et al., 2024; 2026; Zhang et al., 2025). We connect these lines by jointly varying architecture and regime, measuring exact feasibility, and testing whether a shared readout of local stability transfers across instances. ANYCSP (Tönshoff et al., 2023) serves only as evidence that the target instances remain solvable; Appendix E details the probe controls.

3 Methodology

Our experiments instantiate the three axes of the Introduction with three linked measurements. The architecture hierarchy varies what computation the solver performs. Generator-defined regime sweeps vary the difficulty of the instances presented to it. Behavioral evaluation records whether each trained model approximates or solves those instances, while representation analysis asks what

solver-relevant information its final states organize. Applying these measurements to the same model family and regime sweeps lets us relate *which* architecture succeeds, *when* it succeeds, and *what* its representations encode.

3.1 PROBLEMS, OBJECTIVES, AND SUCCESS CRITERIA

We represent every task by decision variables V, a finite label set $[q]$, and local constraints $\mathcal{C}$ whose scores sum to a global objective. SAT and NAE-SAT use $q = 2$ and arity-three clauses; graph coloring uses $q = 3$ and not-equal edge constraints; and Max-Cut uses the same predicate with $q = 2$. We report constraint-satisfaction quality and exact feasibility for SAT, NAE-SAT, and coloring. Every binary labeling is a valid Max-Cut partition, so generic graphs receive objective scores only; for planted two-community graphs, whose cross- and within-block edge probabilities are $2b/n$ and $2a/n$, we additionally report attainment of the planted cut. At $a = 0$ the graph is bipartite and this normalized reference value is one.

3.2 ARCHITECTURE AXIS: THE CUMULATIVE OPTGNN HIERARCHY

The architecture axis is a cumulative hierarchy derived from the relaxation-based OPTGNN (Yau et al., 2024). The four configurations successively add tied iterative refinement, an output trained to penalize relaxed constraint violations, and explicit communication from constraints back to their variables. Adjacent configurations therefore expose what changes when one design choice is added, while the crossed ablation in Section 4 tests the two components highlighted in the Introduction outside the cumulative ordering. Figure 2, Table 1, and Appendix B provide the complete inventory. The historical name OPTGNN-CLAUSE is retained across tasks, but its aggregation operation is problem-general: the local constraint is a clause in SAT/NAE-SAT and an edge in coloring/Max-Cut.

Table 1: Cumulative architecture hierarchy with SAT-configuration budgets. "Constraint messages" are clause-to-variable messages for SAT/NAE-SAT and edge-to-vertex messages for coloring/Max-Cut. Parameters are for hidden width 128; iterations are inference rounds at evaluation; "Cand." is the number of candidate assignments decoded per formula at each iteration ($-$: deterministic decoding). Parameter count is not monotone across variants.

Variant	Update map	Per-step input	Training loss	Decoder / messages	Params	Iter.	Cand.
OPTGNN	32 untied	∇L_{SDP}	L_{SDP}	randomized geometric	2.10M	32	10
OPTGNN-IR	shared	∇L_{SDP}	L_{SDP}	randomized geometric	65.7K	500	10
OPTGNN-LOGIT	shared	∇L_{SDP}	$\sum_C u_C$	threshold / argmax	82.3K	500	–
OPTGNN-CLAUSE	shared	$\nabla L_{\mathrm{SDP}}, \sum_{C \ni v} m_C$	$\sum_C u_C$	threshold / argmax; constraint msgs.	161K	500	–

Base model. Our starting point is OPTGNN Yau et al. (2024), a graph neural architecture coupled to a continuous SDP-inspired relaxation. Given a problem instance, the model maintains a vector representation for each variable, updates these representations using gradients of the relaxation objective, and obtains the final assignment by thresholding the final variable states:

$$h_v^{(t+1)} = \mathrm{Norm}\Big(M_t\Big(h_v^{(t)}, \mathrm{Norm}(\nabla_{h_v} L_{\mathrm{SDP}})\Big)\Big), \qquad \widehat{y}_v = \mathrm{sign}(h_v^{(T)} \cdot r). \tag{1}$$

Here $h_v^{(t)}$ is variable v's state at iteration t, L_{SDP} is the task's SDP-inspired relaxation, and M_t is the learned update map. The original OPTGNN uses 32 untied maps $M_1, \ldots, M_{32}$. Equation (1) shows the binary randomized-hyperplane decoder; coloring uses the corresponding simplex decoder. Decoding budgets and iteration selection are specified once in Section 3's evaluation protocol.

Cumulative modifications. Starting from OPTGNN, each modification changes a distinct part of the computation: how long the update rule can be reused, where the training objective and decoded assignment live, and what local information reaches each variable.

- **Iterative refinement (OPTGNN-IR):** we replace the original sequence of step-specific maps $M_1, \ldots, M_{32}$ with a single shared recurrent map M, which is applied for 32 iterations during training, and up to 500 in inference. This introduces repeated computation with

tied parameters, the same weight-tying principle used to depth-generalize sequence models (Dehghani et al., 2019).

- **Relaxed-label output/loss configuration (OPTGNN-LOGIT):** the update still receives ∇L_{SDP}, but supervision and decoding move together to relaxed labels: sigmoid logits for binary tasks and a softmax for coloring. The loss sums task-specific probabilities of violating each local constraint; for SAT this is the product of the three literal-violation probabilities. NAE-SAT adds the complementary all-equal event, while coloring and Max-Cut use edge-violation probabilities. Full formulas and the graph-task anti-collapse regularizer appear in Appendices H, K, and J. OPTGNN and OPTGNN-IR retain L_{SDP} as both per-step signal and training loss.
- **Constraint-level aggregation (OPTGNN-CLAUSE):** the final modification supplies each variable with an explicit summary of the constraints in which it participates. For SAT/NAE-SAT let $s_{Cv} \in \{+1, -1\}$ denote the polarity of variable v in constraint C. We form polarity-conditioned literal states, a constraint representation, and separate positive- and negative-literal aggregates for each variable,

$$\begin{aligned} \widetilde{h}_{Cv} &= M_{\mathrm{lit}}(h_v, s_{Cv}), & m_C &= \mathrm{Norm}\left(M_C\left(\sum_{v \in C} \widetilde{h}_{Cv}\right)\right), \\ a_v^+ &= \sum_{C \ni v : s_{Cv} = +1} m_C, & a_v^- &= \sum_{C \ni v : s_{Cv} = -1} m_C, \\ h_v^{(t+1)} &= \mathrm{Norm}\Big(M_V\left(h_v^{(t)}, \nabla_{h_v} L_{\mathrm{SDP}}, a_v^+, a_v^-\right)\Big). \end{aligned} \tag{2}$$

For unsigned edge constraints, Equation (2) reduces to the single edge-message aggregate in Eq. (6). Each model is trained separately on its own task. Appendix E uses the exact SAT message concatenation as a matched-input control.

3.3 REGIME AXIS: INSTANCE GENERATION AND DIFFICULTY COHORTS

Random generators let us vary difficulty in a controlled way: despite worst-case NP-hardness, their average-case structure changes systematically with the generating parameters. All synthetic experiments follow the same template: choose a task, size, and regime parameter; sample an instance from the corresponding generator; train on fresh or independently generated instances; and evaluate on disjoint instances, usually at larger sizes and across a wider regime range. The regime parameter is task-specific—clause density for SAT/NAE-SAT, edge density for coloring, and graph or detectability parameters for Max-Cut—and its numerical value is never compared across problems.

Regimes. We define regimes from the instance generator, before observing any neural model. For the running 3-SAT example, the clause density $c = m/n$ yields four descriptive cohorts: sparse ($c \leq 1$), moderate ($c = 2$–3), constrained ($c = 3.5$–3.7), and a near-threshold boundary ($c = 3.9$). These refine the Introduction's three-part description: sparse and moderate form the easy end of the sweep, the constrained cohort is the intermediate regime, and the boundary cohort is the critical end examined here. The labels refer to increasing typical coordination demands under the random generator; they are not new complexity classes or claims about every instance at a given density. Random CSPs are useful precisely because their average-case structure changes with such generator parameters even though the underlying decision problems are NP-hard (Mézard & Zecchina, 2002; Achlioptas & Ricci-Tersenghi, 2006; Achlioptas et al., 2005). NAE-SAT, coloring, and Max-Cut use their own generator parameters and task-specific density scales.

SAT and NAE-SAT. For SAT, we follow the data generation procedure of Yau et al. (2024). Random 3-SAT instances are generated by sampling $m = cn$ clauses uniformly, each containing three randomly selected variables with independently sampled literal signs. Training draws fresh batches at $n = 100$ and $c \sim \mathcal{U}[3.5, 3.9]$, near the satisfiability phase transition at $c \approx 4.26$ Achlioptas et al. (2005); Friedgut et al. (1999). The controlled evaluation uses larger random instances with $n = 1000$ and $c \in \{0.5, 1.0, 2.0, 2.5, 3.0, 3.5, 3.7, 3.9\}$. We stop at $c = 3.9$, below the asymptotic threshold; auxiliary certification at $n = 250$ finds all sampled formulas satisfiable at $c = 3.86$ and 3.9 (Appendix C).

For NAE-SAT, we generate random instances analogously, but evaluate clauses under the not-all-equal constraint. The recurrent architecture and constraint aggregation are unchanged; only the relaxed-label violation term changes (Appendix H). Training uses $n = 100$ and $c \in \{0.5, 1.0, 1.5, 1.8, 2.0, 2.1, 2.5\}$. Evaluation uses disjoint $n = 1000$ formulas; because the within-instance, transfer, and solve-rate analyses use different verified subsets, their exact grids are stated in the corresponding table captions. The highest-density points approach or exceed the empirical NAE satisfiability boundary, so we use them as boundary controls rather than attributing every failure there to solver difficulty.

To check whether our findings transfer beyond synthetically generated instances, we also evaluate on SATLIB Hoos & Stützle (2000), a standard public repository of SAT benchmark instances: specifically, 13 structured, non-random, largely real-world-derived families spanning graph colouring, planning (Blocksworld, Logistics), circuit fault diagnosis (SSA), quasigroup completion, parity learning, inductive inference, Towers of Hanoi, and unsatisfiable combinatorial instances (Appendix I); we do not evaluate SATLIB's random benchmarks, since our own synthetic data (Section 4) already covers the uniform-random regime.

Graph coloring and Max-Cut. Coloring uses planted proper 3-colorings at $n = 1000$ over edge density $c \in [1, 20]$, with size-shift checks. Max-Cut uses four synthetic graph families, four TU datasets (Morris et al., 2020), and a planted two-community family with a known achievable cut, evaluated at $n \in \{200, 400, 800\}$. Training and evaluation graphs are disjoint; generator parameters, split sizes, and schedules appear in Appendices K and J.

3.4 Behavioral evaluation

Behavioral evaluation answers whether a trained model succeeds in each regime. Quality is the satisfied-clause fraction for SAT/NAE-SAT, properly colored-edge fraction for coloring, and cut value for Max-Cut. Success means satisfying every constraint for the first three tasks and attaining the planted reference (within .005) for planted Max-Cut; the *solve/attainment rate* is the fraction of held-out instances meeting that criterion. Generic Max-Cut graphs and UNSAT-by-construction SATLIB families receive quality scores only.

The evaluated checkpoints use hidden dimension 128 and a fixed task-specific training budget shared by all variants within each comparison. SAT/NAE-SAT use two epochs of 2,000 batches per epoch (batch size 8, Adam, learning rate 2×10^{-4}); graph-task schedules appear in their appendices. Inference uses 32 iterations for OptGNN and up to 500 for the recurrent variants, so broad comparisons characterize complete solver configurations under their stated decoder and search budgets. The adjacent OptGNN-Logit–OptGNN-Clause comparison that isolates constraint aggregation is budget-matched. We decode and score every iteration, retaining the earliest iteration with the best objective quality. For SAT this is the first iteration whose assignment satisfies every clause, a check that any solver can apply to its own output (a verifier over the given formula, never the planted or a reference solution), applied identically to all variants; solve rates therefore measure success within the stated iteration budget. SDP-style binary models use 10 randomized-hyperplane samples and retain the best decoded assignment; coloring uses simplex decoding. Logit models use thresholding for binary tasks and argmax for coloring. A fixed reference draw is used only for trajectory comparisons such as flip dynamics. Appendix C audits rounding-budget sensitivity. The retained assignment is also the one analyzed below.

For failed SAT assignments, a secondary diagnostic uses CP-SAT with a 30-second budget to obtain an upper bound on normalized distance to the nearest satisfying assignment. Because this optimization is expensive, it uses separate $n = 250$, $c = 3.7$ instances; the corresponding results appear in Section 4.

3.5 Concept axis: representation analysis

Behavioral evaluation establishes whether and when a solver configuration succeeds; the concept axis asks which task-derived statistic is linearly accessible from its final hidden states. We hold a trained model fixed, take its selected assignment ϕ, compute a target $g_\phi(v)$ for every variable or vertex, and ask whether that target is recoverable by a linear readout trained on some instances and evaluated on held-out ones. Its relationship with solving is established by comparing accessibility and behavior across architectures and regimes.

Task-derived measures. SAT uses unique-clause support. NAE-SAT uses odd-one-out support: the number of clauses in which a variable's literal is the only one taking its value, so that flipping it would make the clause all equal. For coloring and Max-Cut we use the signed edge margin

$$g_\phi(v) = \#\{u \sim v : \phi(u) \neq \phi(v)\} - \#\{u \sim v : \phi(u) = \phi(v)\}.$$

For Max-Cut the signed margin is the objective loss from flipping v and therefore measures local stability directly. For $q = 3$ coloring it is a current-color agreement margin; the cost of recoloring depends on which alternative color is chosen. We treat the coloring statistic as an analogous local diagnostic, without identifying it with SAT support.

Let $\mathbf{H} \in \mathbb{R}^{n \times d}$ denote the final variable-state matrix. We compute g_ϕ under the model's own assignment selected by the behavioral protocol: the best of the 10 randomized geometric decodes for OPTGNN/OPTGNN-IR, and threshold or coloring argmax for OPTGNN-LOGIT/OPTGNN-CLAUSE.

Probe protocols. The primary *cross-instance transfer* protocol tests the reusable encoding invoked in the Introduction. One linear probe is fitted to nodes pooled from 20 training instances and applied unchanged to 20 disjoint instances generated with separate seeds. SAT and NAE-SAT use a linear SVM for each adjacent support pair $(s, s+1)$, $s \in \{0, 1, 2\}$, and report held-out Macro-F1 for each pair. Coloring and Max-Cut use ridge regression on the integer-valued margin and report held-out R^2. The secondary *within-instance* protocol performs five-fold cross-validation over the nodes of each held-out instance and then averages across instances. For the common-reference SAT control, we additionally draw planted-satisfiable formulas at matched n and c: a hidden satisfying assignment is sampled first and clauses violating it are rejected, so every architecture's support target is computed from the same known assignment. The result captions identify which protocol and baselines apply to each table.

Update-dynamics diagnostics condition assignment changes on current support or margin; their definitions and results appear in Appendix E.

4 RESULTS

The Introduction posed a paradox: closely related solvers can be nearly indistinguishable by approximation while differing completely in exact success. We resolve it through three empirical questions, corresponding to the three axes of our framework. First, *where* across the difficulty regimes do approximation and exact success diverge? Second, *which* architectural components account for that behavioral separation? Third, is local stability organized in the models' representations, and how does its relationship with solving change across regimes and problems? We answer each question using 3-SAT as the running example, then test the scope of the resulting account on NAE-SAT, 3-coloring, and Max-Cut.

4.1 APPROXIMATION DOES NOT PREDICT EXACT SUCCESS, AND THE SEPARATION DEPENDS ON REGIME

Across the broader sweep from $c = 1$ onward, all four SAT variants satisfy at least .967 of clauses; at the primary density $c = 3.7$, the range is .970–1.000 after rounding, yet their feasibility rates diverge sharply (Table 2). The separation emerges as the instances leave the easy regime. At intermediate difficulty, OPTGNN and OPTGNN-IR cease solving, while OPTGNN-LOGIT becomes sensitive to the training run; recurrence or the relaxed-label output alone therefore does not robustly close the approximation–feasibility gap. We measure this at two scales. A three-model evaluation uses 30 formulas per model, whereas the separate probe-matched evaluation uses four independently trained models and 100 formulas per model. These are separate samples rather than nested averages. In the probe-matched evaluation, OPTGNN-CLAUSE solves $254/400 = .635$ formulas at $c = 3.5$ and $142/400 = .355$ at $c = 3.7$; every other variant remains at or below .003 at the former density and at zero at the latter. At the critical end of our sweep ($c = 3.9$), all variants retain near-perfect approximation and none solves.

NAE-SAT reproduces the same ordering on its own density scale (Appendix H). OPTGNN-CLAUSE-NAE remains the strongest solver into intermediate difficulty, while OPTGNN-IR fails and

Table 2: Random SAT at $n = 1000$. Each cell reports exact-feasibility rate (mean $\pm$ between-model SD) | mean satisfied-clause ratio; larger is better for both metrics. The first four rows summarize three independently trained models evaluated on 30 held-out formulas each. The † row reports the separate four-model, 100-formula evaluation used for the representation analyses; rates across the two evaluations are not paired estimates. Pooled counts appear in Table 3; Appendix C reports robustness checks.

Model	$c = 0.5$	$c = 1.0$	$c = 2.0$	$c = 2.5$
OPTGNN	$.02 \pm .04$ | .923	$.06 \pm .04$ | .995	$.00 \pm .00$ | .985	$.00 \pm .00$ | .978
OPTGNN-IR	$.69 \pm .07$ | .999	$.04 \pm .02$ | .996	$.00 \pm .00$ | .990	$.00 \pm .00$ | .986
OPTGNN-LOGIT	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000	$.71 \pm .39$ | 1.000	$.41 \pm .48$ | .999
OPTGNN-CLAUSE	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000
OPTGNN-CLAUSE†	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000	$1.00 \pm .00$ | 1.000

Model	$c = 3.0$	$c = 3.5$	$c = 3.7$	$c = 3.9$
OPTGNN	$.00 \pm .00$ | .976	$.00 \pm .00$ | .972	$.00 \pm .00$ | .970	$.00 \pm .00$ | .967
OPTGNN-IR	$.00 \pm .00$ | .982	$.00 \pm .00$ | .978	$.00 \pm .00$ | .977	$.00 \pm .00$ | .976
OPTGNN-LOGIT	$.19 \pm .25$ | .999	$.00 \pm .00$ | .998	$.00 \pm .00$ | .998	$.00 \pm .00$ | .997
OPTGNN-CLAUSE	$.99 \pm .02$ | 1.000	$.79 \pm .19$ | 1.000	$.49 \pm .25$ | 1.000	$.00 \pm .00$ | .999
OPTGNN-CLAUSE†	$.99 \pm .02$ | 1.000	$.64 \pm .21$ | 1.000	$.36 \pm .25$ | 1.000	$.00 \pm .00$ | .999

OPTGNN-LOGIT collapses earlier. This decline is not explained by unsatisfiable inputs: auxiliary CP-SAT certification finds every sampled $n = 250$ formula satisfiable through $c = 1.9$ and $28/30$ satisfiable at $c = 2.0$; at $n = 1000$, all five spot checks at both densities are satisfiable. The decline beginning at $c = 1.75$ therefore reflects solver behavior. The $c = 2.1$ cohort mixes satisfiable and unsatisfiable formulas and serves only as a boundary control. Across both satisfiability tasks, near-perfect approximation can coexist with sharply different feasibility.

Near-perfect approximation also need not place a valid solution one trivial repair away: at $n = 250, c = 3.7$, CP-SAT finds satisfying assignments within 17% and 4.4% of failed OPTGNN and OPTGNN-CLAUSE outputs, respectively. These are upper bounds on nearest-solution distance under a 30-second budget. Yet WalkSAT and Survey Propagation solve every tested instance through $c = 3.7$, and ANYCSP (Tönshoff et al., 2023) solves all tested instances through $c = 3.9$. The intermediate regime therefore exposes a separation within these evaluated solver configurations rather than an intrinsic hardness boundary; Appendix C gives certification and solver-calibration details.

4.2 NEITHER COMPONENT ALONE CLOSES THE GAP; THEIR COMBINATION DOES

The cumulative hierarchy alone cannot identify which addition creates the separation. We therefore cross constraint aggregation with the relaxed-label training objective and readout at the intermediate-regime densities. At $c = 3.7$, the configuration with neither component, the logit configuration without aggregation (pooled over four independently trained models), and a newly trained aggregation-plus-SDP configuration (three seeds, each 0%) each solve 0%; the full configuration solves 35.5% ($142/400$; every trained Clause model solves at both densities; per-model counts are in Appendix Table 3). At $c = 3.5$, the corresponding rates are 0%, $.3\%$, 0%, and 63.5%. A separate check retrained the logit-only configuration under a non-symmetric initialization (§C); it still solves 0% at both $c = 3.5$ and $c = 3.7$. This factorial comparison is consistent with a configuration-level interaction: neither tested component alone reproduces the combined configuration's feasibility gain. Appendix A reports the factorial cells, model counts, and initialization control together (Table 4).

4.3 LOCAL STABILITY TRACKS ARCHITECTURE RATHER THAN SOLVING

Architecture-level accessibility. At $c = 3.7$, a common satisfying assignment defines labels for every architecture. Across the three adjacent-support tasks in Figure 1(d), the three non-solving architectures yield Macro-F1 .49–.52 over four independently trained models (SD $\leq$ .02), near chance, whereas OPTGNN-CLAUSE reaches .576–.599. Aggregation with the SDP objective solves nothing but reaches nearly the same range, .564–.590, over three independently trained models. Support accessibility therefore follows constraint aggregation rather than exact solving.

Controls and provenance of the signal. Permuted labels yield Macro-F1 .49–.50 (maximum .52), below every OPTGNN-CLAUSE score (.56–.60). More importantly, the trained SAT and NAE-SAT states exceed both the exact incoming aggregate and its combination with assignment statistics by .07–.08 Macro-F1; matched-dimension and random-weight controls agree (Appendix Table 5). The signal is thus more than a linear copy of these tested inputs and is largely present before the selected update. In coloring the aggregate already matches the state; Max-Cut is inconclusive.

Instance-level geometry. Distance to the best satisfying assignment found by CP-SAT within 30 seconds separates solved from failed OPTGNN-CLAUSE instances with AUROC $\geq .99$, whereas support from the model's own decode does not. Removing the probe direction behaves like removing a random direction, and its repair rule solves 4% of failures versus 28% for min-break (Appendix G).

Across-regime interpretation. The configuration ordering persists at $c = 3.0, 3.7, 3.9$ (Figure 1), but OPTGNN-CLAUSE's accessibility remains nearly unchanged as its solve rate falls to zero; the complete NAE-SAT density sweep shows the same separation (Appendix Table 8). Accessibility distinguishes configurations in the intermediate regime but does not track success across regimes: local stability remains organized after exact success disappears.

4.4 ARCHITECTURE TRANSFERS ACROSS PROBLEMS; REPRESENTATION REMAINS TASK-DEPENDENT

The other problems separate what transfers robustly from what depends on the task. The architectural signature is stable under the stated budgets: once difficulty becomes informative, the configuration combining aggregation with the relaxed-label output/loss is strongest in NAE-SAT and the planted coloring family, and retains a target-attainment advantage on planted Max-Cut near the detectability boundary. The representational account is more specific. In SAT and NAE-SAT, trained states organize support beyond the tested inputs and the encoding outlasts solving. In coloring, the learned incoming aggregate already carries the edge margin, so the final state adds no measurable accessibility; on generic Max-Cut, where every labeling is feasible, accessibility decouples from cut quality. Across problems, the architecture-level separation is the robust finding, while the provenance and behavioral relevance of the probed concept depend on the constraint system and regime (Appendices H, K, and J).

5 DISCUSSION AND CONCLUSION

The approximation–feasibility gap is a failure of evaluation as much as a failure of optimization. Objective quality varies smoothly across our hierarchy, whereas feasibility exposes a sharp architectural separation and the regime in which it appears. A single aggregate score would miss both facts. The three axes play distinct roles: architecture affects which task-relevant quantities are linearly accessible from the representation; the represented concept reveals what information the solver has made readily usable; and the instance regime determines whether that information is enough to complete a valid assignment. Their intersection explains why closely related models can look equivalent by approximation and behave differently as solvers.

The crossed ablation sharpens the design implication. Constraint aggregation makes local stability transferable across SAT and NAE-SAT instances, including in a configuration that never solves the primary cohort. Feasible solving emerges only when aggregation is combined with the relaxed-label output/loss configuration in the recurrent architecture. This separates two design questions that objective value conflates: whether the representation organizes a useful algorithmic quantity, and whether the remaining update and decoding pathway can exploit it at the target difficulty. Probes can therefore serve as development diagnostics. A new architecture that improves feasibility without changing the tested concept suggests a different computational advantage; one that strengthens the concept without improving feasibility identifies information that is present but still underused.

These findings suggest a concrete evaluation standard for neural combinatorial solvers. Unless approximation is the intended endpoint, comparisons should pair objective quality with exact feasibility or target attainment, span generator-defined regimes, report decoding and search budgets, and test representation claims on new instances.

AI USE STATEMENT

We used generative AI tools for wording and structural editing and to simulate reviewer feedback. The authors reviewed and verified the final text and take responsibility for all claims, code, figures, and analyses.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the architecture hierarchy, data-generating processes, evaluation budgets, success criteria, and representation-analysis protocol. The appendices report per-model counts, decoding and initialization checks, satisfiability certification, probe controls, complete task-specific results, software, and compute details. The anonymized archive will provide code, configurations, checkpoints, seeds, and primary-result data. Four independently trained models are used in each probe-matched SAT and NAE-SAT analysis. Unless a caption states otherwise, the behavioral coloring and Max-Cut sweeps use one trained checkpoint per variant; the four-model raw-message controls are reported separately. Training takes under one hour per model on one RTX 3090/3080 GPU.

REFERENCES

Dimitris Achlioptas and Amin Coja-Oghlan. Algorithmic barriers from phase transitions. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pp. 793–802. IEEE Computer Society, 2008. doi: 10.1109/FOCS.2008.11. URL https://doi.org/10.1109/FOCS.2008.11.

Dimitris Achlioptas and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In *Proceedings of the 38th ACM Symposium on Theory of Computing*, pp. 130–139, 2006.

Dimitris Achlioptas, Arthur D. Chtcherba, Gabriel Istrate, and Cristopher Moore. The phase transition in 1-in-k SAT and NAE 3-sat. In S. Rao Kosaraju (ed.), *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pp. 721–722. ACM/SIAM, 2001. URL http://dl.acm.org/citation.cfm?id=365411.365760.

Dimitris Achlioptas, Assaf Naor, and Yuval Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, 2005.

Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs (preliminary version). In *Proceedings of the twenty-sixth annual ACM symposium on Theory of Computing*, pp. 346–355, 1994.

Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733–1748, 1997.

Victor Bapst, Amin Coja-Oghlan, Samuel Hetterich, Felicia Raßmann, and Dan Vilenchik. The condensation phase transition in random graph coloring. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore (eds.), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, Barcelona, Spain, September 4-6, 2014*, volume 28 of *LIPIcs*, pp. 449–464. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014. doi: 10.4230/LIPICS.APPROX-RANDOM.2014.449. URL https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.449.

Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman (eds.), *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pp. 151–158. ACM, 1971. doi: 10.1145/800157.805047. URL https://doi.org/10.1145/800157.805047.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6): 066106, 2011.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations (ICLR)*, 2019.

Abraham Flaxman. A spectral technique for random satisfiable 3cnf formulas. *Random Struct. Algorithms*, 32 (4):519–534, 2008. doi: 10.1002/RSA.20213. URL https://doi.org/10.1002/rsa.20213.

Ehud Friedgut, Jean Bourgain, et al. Sharp thresholds of graph properties, and the k-SAT problem. *Journal of the American Mathematical Society*, 12(4):1017–1054, 1999.

Justin et al. Gilmer. Neural message passing for quantum chemistry. In *ICML*, 2017.

Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

Holger H. Hoos and Thomas Stützle. Local search algorithms for SAT: an empirical evaluation. *J. Autom. Reason.*, 24(4):421–481, 2000. doi: 10.1023/A:1006350622830. URL https://doi.org/10.1023/A:1006350622830.

Holger H Hoos and Thomas Stützle. Satlib: An online resource for research on sat. *Sat*, 2000:283–292, 2000.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher (eds.), *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pp. 85–103. Plenum Press, New York, 1972. doi: 10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems*, 30, 2017.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Henrique Lemos, Marcelo O. R. Prates, Pedro H. C. Avelar, and Luís C. Lamb. Graph colouring meets deep learning: Effective graph neural network models for combinatorial problems. *CoRR*, abs/1903.04598, 2019. URL http://arxiv.org/abs/1903.04598.

Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. *Advances in neural information processing systems*, 31, 2018.

Andrea Lincoln and Adam Yedidia. Faster random k-cnf satisfiability. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli (eds.), *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pp. 78:1–78:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ICALP.2020.78. URL https://doi.org/10.4230/LIPIcs.ICALP.2020.78.

Marc Mézard and Riccardo Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Physical Review E*, 66(5):056126, 2002.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Eugene Nudelman, Kevin Leyton-Brown, Holger H Hoos, Alex Devkar, and Yoav Shoham. Understanding random sat: Beyond the clauses-to-variables ratio. In *Principles and Practice of Constraint Programming–CP 2004: 10th International Conference, CP 2004, Toronto, Canada, September 27-October 1, 2004. Proceedings 10*, pp. 438–452. Springer, 2004.

Bart Selman, Henry A. Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In David S. Johnson and Michael A. Trick (eds.), *Cliques, Coloring, and Satisfiability, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, October 11-13, 1993*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 521–531. DIMACS/AMS, 1993. doi: 10.1090/dimacs/026/25. URL https://doi.org/10.1090/dimacs/026/25.

Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. Learning a SAT solver from single-bit supervision. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJMC_iA5tm.

Guilhem Semerjian and Rémi Monasson. Relaxation and metastability in a local search procedure for the random satisfiability problem. *Physical Review E*, 67(6):066103, 2003.

Elad Shoham, Hadar Cohen, Khalil Wattad, Havana Rika, and Dan Vilenchik. Concept learning in the wild: Towards algorithmic understanding of neural networks, 2024.

Elad Shoham, Omri Haber, Havana Rika, and Dan Vilenchik. Learning to rank: How gnns solve max-clique and sparse PCA. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 25401–25409. AAAI Press, 2026. doi: 10.1609/AAAI.V40I30.39734. URL https://doi.org/10.1609/aaai.v40i30.39734.

Jan Tönshoff, Martin Ritzert, Hinrikus Wolf, and Martin Grohe. Graph Neural Networks for Maximum Constraint Satisfaction. *Frontiers in Artificial Intelligence*, 3, 2021.

Jan Tönshoff, Berke Kisin, Jakob Lindner, and Martin Grohe. One model, any CSP: Graph neural networks as fast global search heuristics for constraint satisfaction. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

Dan Vilenchik. It's all about the support: a new perspective on the satisfiability problem. *Journal on Satisfiability, Boolean Modeling, and Computation*, 3(3-4):125–139, 2007.

Morris Yau, Nikolaos Karalias, Eric Lu, Jessica Xu, and Stefanie Jegelka. Are graph neural networks optimal approximation algorithms? In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/85db52cc08c5e00cfb1d216b1c85ba35-Abstract-Conference.html.

Zhiqin Zhang, Yining Ma, Zhiguang Cao, and Hoong Chuin Lau. Probing neural combinatorial optimization models, 2025. URL https://arxiv.org/abs/2510.22131.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A SUPPLEMENTARY PRIMARY-RESULT EVIDENCE

Table 3: Exact-solve counts for each independently trained model at the primary SAT densities ($n = 1000$, 100 formulas per model). Pooled OPTGNN-CLAUSE rates are $254/400 = .635$ and $142/400 = .355$.

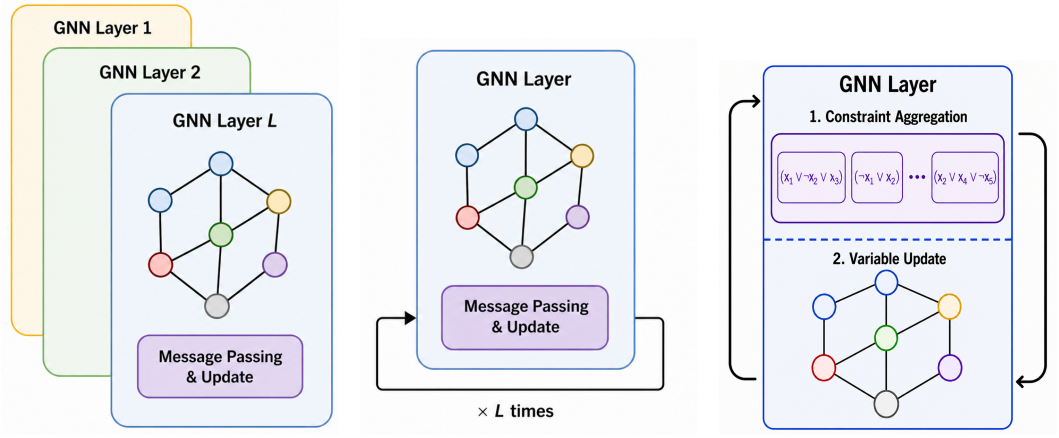
c	Variant	Model 1	Model 2	Model 3	Model 4
3.5	OPTGNN-CLAUSE	39	81	82	52
3.5	OPTGNN-LOGIT	0	1	0	0
3.5	OPTGNN, OPTGNN-IR	0	0	0	0
3.7	OPTGNN-CLAUSE	11	55	59	17
3.7	other three variants	0	0	0	0

Table 4: Crossed SAT configuration ablation at the two intermediate-regime densities. "Aggregation" denotes explicit constraint-to-variable communication and "relaxed-label output/loss" denotes the jointly changed readout, decoder, and constraint-penalizing loss. The first, second, and fourth rows use the probe-matched four-model evaluation (100 held-out formulas per model); the aggregation-plus-SDP row reports zero solves in each of three independently trained models. The final row is a separately retrained initialization control.

Configuration	Models	$c = 3.5$	$c = 3.7$
Neither component	4	$0/400$	$0/400$
Relaxed-label output/loss only	4	$1/400$	$0/400$
Aggregation plus SDP objective	3	0% in each model	0% in each model
Both components	4	$254/400$	$142/400$
Relaxed-label output/loss only, non-symmetric init.	1	0%	0%

Table 5: Cross-instance accessibility of the trained final state relative to controls: H_{post} minus the named control (difference [95% hierarchical-bootstrap interval]). SAT/NAE-SAT use Macro-F1; graph tasks use R^2.

Problem	A_{raw}	B	$B + A_{\text{raw}}$	H_{pre}
SAT	$+.081$ $[+.066, +.096]$	$+.326$ $[+.299, +.354]$	$+.073$ $[+.059, +.087]$	$+.009$ $[+.007, +.011]$
NAE-SAT	$+.076$ $[+.071, +.083]$	$+.237$ $[+.228, +.245]$	$+.071$ $[+.065, +.077]$	$+.007$ $[+.005, +.010]$
3-coloring	$-.004$ $[-.013, +.006]$	$-.024$ $[-.032, -.015]$	$-.040$ $[-.045, -.035]$	$+.002$ $[+.0003, +.004]$
Max-Cut	$-.021$ $[-.051, +.039]$	$+.130$ $[+.017, +.262]$	$-.031$ $[-.063, +.034]$	$-.011$ $[-.052, +.020]$



(a) Stacked architecture. (b) Shared recurrent layer. (c) Clause-level aggregation.

(d) Logit-space readout and loss.

Figure 2: Inventory of the cumulative OPTGNN progression. The panels document how recurrence, logit readout and loss, and constraint aggregation enter the compared systems.

B ARCHITECTURE INVENTORY

Figure 2 below diagrams the four compared systems described in Section 3: the stacked base OPTGNN, the shared-recurrent-layer OPTGNN-IR, the logit-space readout and loss of OPTGNN-LOGIT, and OPTGNN-CLAUSE's added clause-level aggregation.

C DECODING, INITIALIZATION, AND SATISFIABILITY CHECKS FOR THE SAT RESULTS

Table 2 reports one matched evaluation for every variant. This appendix reports how those results depend on decoding budget, initialization, and instance satisfiability.

Random-hyperplane decoding. The SDP-style OPTGNN and OPTGNN-IR variants depend on randomized hyperplane rounding, whereas the logit variants decode directly. The matched evaluation in Table 2 uses ten hyperplane samples (Section 3). Holding the checkpoint and instances fixed while increasing only this budget raised OPTGNN at $c = 0.5$ to 10/10 solved with 10^6 samples and OPTGNN-IR to 1.00 with 100 samples. Their sparse-regime solving rates therefore characterize the joint model–decoder system at the stated budget. At $c = 3.5$, increasing the budget from 10 to 10^5 produced no solutions for either variant on 30 held-out instances in each of two independently trained models ($0/30$ in every cell; Clopper–Pearson upper bound .116), and at $c = 3.7$ none on five held-out instances per cell, bounding this explanation at the densities used for the architectural comparison. Random rounding also injects noise into the assignment-relative support target used by the probe. Replacing one stochastic decode with a 25-decode majority target changes SAT adjacent-pair Macro-F1 only from approximately .50 to .51–.53; the corresponding NAE-SAT values remain .49–.51. Thus stochastic decoding is a real measurement asymmetry but does not explain the large Clause probe advantage under this robustness check.

Logit initialization. OPTGNN-LOGIT initializes every variable at the symmetric state. An alternative non-symmetric initialization solves .97 and .93 of the instances at $c = 2.5$ in two seeds, and leaves the result at the crossed-ablation densities unchanged: a fresh seed solves 0% at both $c = 3.5$ and $c = 3.7$.

Instance satisfiability near the boundary. Auxiliary CP-SAT checks certify all 30 sampled random SAT formulas at $n = 250$, $c \in \{3.86, 3.9\}$. For NAE-SAT, all 30 sampled formulas are satisfiable through $c = 1.9$, $28/30$ at $c = 2.0$, and only $11/30$ at $c = 2.1$; five $n = 1000$ spot checks are satisfiable at both $c = 1.9$ and $c = 2.0$, while $c = 2.1$ times out. These are auxiliary samples rather than certificates for every headline instance. They support interpreting the earlier NAE collapse as model failure while requiring explicit qualification of the $c = 2.1$ boundary.

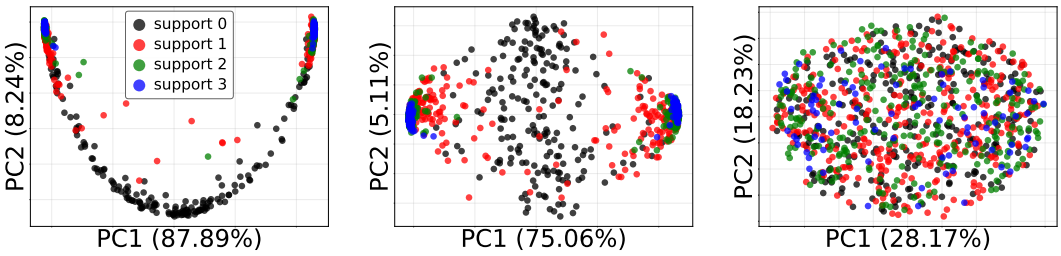
D ADDITIONAL REPRESENTATION AND DYNAMICS ANALYSES

Under a uniformly random Boolean assignment (as at model initialization, before any learned or search-driven bias toward satisfying configurations), for a fixed variable in a 3-literal clause, exactly one of the $2^3 = 8$ local truth assignments makes that variable the unique satisfying literal. Since $m = cn$ clauses contain $3m$ literal occurrences, each variable appears in $3c$ clauses on average. The expected number of support clauses per variable under this uniform baseline is therefore $3c/8$; once the network's iterative refinement biases the assignment toward satisfying configurations, the empirical frequency of local truth patterns shifts and no longer matches this uniform-assignment calculation exactly, so $3c/8$ should be read as a baseline reference point rather than a prediction for the converged assignment. This gives a simple regime distinction: for $c < 8/3 \approx 2.67$, the expected support is below 1, so the signal is sparse; for larger c, support becomes a more frequently expressed property that can organize the learned representations.

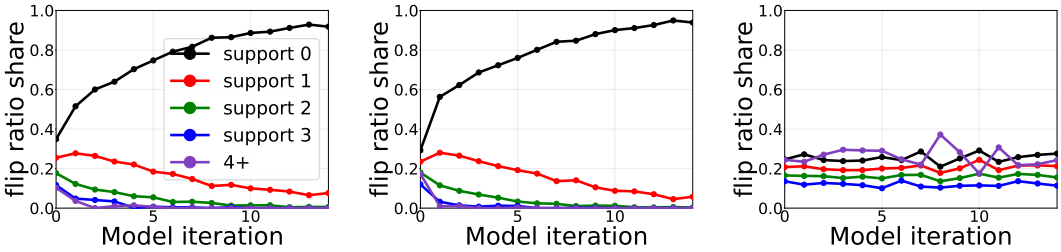
For NAE-SAT, the analogous support event is that the variable is the "odd one out" in its clause: its literal is the only true literal or the only false literal. Under the same uniform-random-assignment baseline as above, for a fixed variable in a 3-literal clause, this occurs in two of the $2^3 = 8$ local truth assignments. Thus, the expected NAE-support per variable under this baseline is $3c \cdot 2/8 = 3c/4$, crossing 1 already at $c = 4/3$. Hence, support is expected to appear at lower densities in NAE-SAT than in SAT.

The cross-instance results in Figure 1(c–d) match this prediction for SAT. OPTGNN and OPTGNN-IR remain near chance across all evaluated densities, while OPTGNN-LOGIT is only slightly above chance and OPTGNN-CLAUSE is consistently strongest. The corresponding NAE-SAT results in Table 7 show a similar qualitative ordering, with support structure appearing at lower densities, consistent with the higher expected support rate in NAE-SAT. Figure 3 provides a qualitative single-checkpoint view of how support appears in PCA space and how assignment changes vary with support.

For Max-Cut, the signed edge margin (Appendix J) is the direct arity-2 analogue of support: it sums whether incident edges agree or disagree with the current cut. Graph coloring uses the same observable as a current-color agreement margin, while the cost of recoloring depends on the



(a) Embedding OPTGNN-CLAUSE (b) Embedding OPTGNN-LOGIT (c) Embedding OPTGNN / OPTGNN-IR



(d) Flip dynamics OPTGNN-CLAUSE (e) Flip dynamics OPTGNN-LOGIT (f) Flip dynamics OPTGNN / OPTGNN-IR

Figure 3: Representation and dynamics across model variants on $n = 1000$, $c = 3.7$ SAT instances. Top row: PCA projections of final variable embeddings colored by support; PC1/PC2 labels include explained variance. Bottom row: exploratory support-conditioned flip dynamics from the displayed checkpoint. OPTGNN and OPTGNN-IR weakly represent support and still flip high-support variables. In this checkpoint, OPTGNN-LOGIT and OPTGNN-CLAUSE show similar support-guided flip dynamics, but OPTGNN-CLAUSE organizes support more clearly in embedding space. These dynamics provide a single-checkpoint illustration.

alternative color selected (Appendix K). These edge statistics have no combinatorial dilution factor: since $\sum_v \deg(v) = 2|E| = 2cn$, the expected degree is $d = 2c$, and every incident edge contributes directly, unlike SAT's $3c/8$ and NAE-SAT's $3c/4$, where only a $1/8$ or $2/8$ fraction of a clause's local truth patterns count at all.

The probe-score ordering also appears in these extensions. OPTGNN-CLAUSE's local-margin statistic is linearly accessible with $R^2 = .45$–$.84$ across all eight Max-Cut datasets, against $R^2 \leq .07$ for OPTGNN and OPTGNN-IR on every one of them (Table 11). Graph coloring shows an even sharper ordering: OPTGNN-CLAUSE's probe is already near its ceiling ($R^2 = .958$) at the lowest density tested ($c = 1$) and stays $\geq .93$ throughout, including where its own SOLVED fraction is 0, while the other three variants stay well below this at low-to-moderate density and reach at most $R^2 = .47$, only past $c \approx 11$ (Table 15). Yet the matched-input control (Appendix E) shows that coloring's raw learned-message aggregate already predicts this margin as well as the final state. Thus a decodable-but-unsolved regime is a boundary for both the solving claim and the stronger learned-state interpretation.

Dynamics: update alignment with local stability. We next ask whether update statistics align with the same local-stability quantity. In SAT, we track which variables flip across iterations as a function of their support. The expected behavior is simple: low-support variables should be revised first, because changing them disrupts few satisfied clauses, whereas high-support variables should be preserved because they uniquely sustain many constraints. This mirrors local-search heuristics such as WalkSAT, which revise unstable variables while avoiding damage to stable parts of the assignment Selman et al. (1993); Hoos & Stützle (2000); Semerjian & Monasson (2003). It is also consistent with Shoham et al. (2024), where support was identified as the guiding motif behind successful neural SAT solving.

In the displayed checkpoint, the flip dynamics in Figure 3 follow the same broad progression as the representation analysis. OPTGNN and OPTGNN-IR behave similarly: both continue to flip high-support variables with relatively high probability, so their update statistics are less aligned with preserving locally stable parts of the assignment. In this exploratory comparison, recurrence alone shows weaker alignment with support, both in representation accessibility and in selective-refinement statistics. By contrast, OPTGNN-LOGIT and OPTGNN-CLAUSE both exhibit the desired support-guided dynamics: flips are concentrated on low-support variables, while high-support variables are largely preserved. Their main difference is representational rather than dynamical. As shown in Figure 1(c–d), OPTGNN-CLAUSE makes support more clearly accessible across unseen instances, whereas Figure 3 suggests that OPTGNN-LOGIT can show a similar support-conditioned flipping pattern despite much weaker transferable organization. Table 7 in the Appendix reports exploratory SVM results for NAE-SAT.

We measure an analogous statistic on Max-Cut and graph coloring: the fraction of nodes that change side/color between consecutive inference steps, binned by absolute-margin tercile, at representative difficult settings (Max-Cut: planted a=4, b=12, n=400; coloring: the low-solve dip at c=2.5). Table 6 does not fully mirror SAT: OPTGNN-CLAUSE shows a low-to-high gap in the same direction on both problems, with the lowest flip rate of any variant at every coloring bin and at high confidence on Max-Cut (OPTGNN-LOGIT is lower at low and medium confidence there). The other three variants are inconsistent in a way that does not reduce to a single "flat vs. decreasing" story: OPTGNN-LOGIT is essentially flat on Max-Cut ($.119 \rightarrow .111 \rightarrow .114$) and only weakly decreasing on coloring; OPTGNN is non-monotonic on Max-Cut (rising from low to medium confidence before falling) while decreasing appreciably on coloring; and OPTGNN-IR, in absolute terms, shows the *largest* low-to-high drop of any variant on Max-Cut ($.445 \rightarrow .318$, a larger absolute change than OPTGNN-CLAUSE's own $.194 \rightarrow .109$) alongside a much smaller one on coloring. Thus, update alignment is problem- and architecture-dependent rather than a general consequence of the logit readout, and the baselines' patterns do not consistently track their overall flip-rate magnitude.

E RAW-MESSAGE AND ASSIGNMENT-CONDITIONED CONTROLS

Question and matched design. The main probe asks whether local stability is linearly accessible from a trained variable state. We additionally trace where that information enters the computation. In particular, OPTGNN-CLAUSE passes learned incident-constraint messages into the variable update,

Table 6: Local-margin-conditioned flip rate (fraction of nodes changing side/color between consecutive inference steps), binned by absolute-margin tercile at representative difficult settings. OPTGNN-CLAUSE shows a low-to-high gap on both problems; the other variants are inconsistent.

Problem	Model	Low conf.	Med. conf.	High conf.
Max-Cut (a=4)	OPTGNN	.442	.452	.404
	OPTGNN-IR	.445	.426	.318
	OPTGNN-LOGIT	.119	.111	.114
	OPTGNN-CLAUSE	**.194**	**.147**	**.109**
Coloring (c=2.5)	OPTGNN	.093	.047	.036
	OPTGNN-IR	.028	.011	.008
	OPTGNN-LOGIT	.170	.155	.143
	OPTGNN-CLAUSE	**.016**	**.007**	**.004**

allowing a direct comparison between input accessibility and post-update organization. We therefore evaluate the exact Clause checkpoints used in the regime analysis without retraining. For SAT, NAE-SAT, planted 3-coloring, and planted Max-Cut, every feature comparison uses the same checkpoint, decoded assignment, target, and 20 training/20 disjoint test instances per setting. The grids are $c \in \{3.5, 3.7, 3.9\}$ for SAT, $c \in \{1.5, 1.75, 1.9\}$ for NAE-SAT, $c \in \{1, 2, 2.5, 3, 4\}$ for coloring, and planted Max-Cut signal $a \in \{0, 4, 6\}$ with $b = 12$. Four independently trained Clause models are tested in each cell, with OPTGNN-LOGIT evaluated on the corresponding instances. The selected hook is the update step used for the model's best decoded output.

Features and safeguards. We capture H_{pre} immediately before an update, each incoming learned constraint message, A_{raw} as the exact positive/negative message aggregates concatenated into that update, and H_{post} after its learned map and normalization. The assignment-conditioned control B includes degree/incidence, SAT literal signs where applicable, the model's decoded output score, and relaxed-loss moments and quantiles; for SAT/NAE-SAT it also includes satisfied/violated incident-constraint summaries. For coloring and Max-Cut we exclude same/different-neighbor counts because those would algebraically reconstruct the local-margin target; the resulting feature set is a non-leaking control, not a deliberately weak target proxy. We also probe $B + A_{\text{raw}}$, pre-update states, gradients, message moments, and scalar relaxed-loss aggregates. All probe normalization and regularization selection use training instances only. Fixed label-free random projections match the dimensions of H_{post} and A_{raw} pairwise (Table 5's dimensionality-matched contrast); the higher-dimensional $B + A_{\text{raw}}$ concatenation is compared to H_{post} at its own, larger dimensionality, without projection, so its contrast in Table 5 is not dimensionality-matched and should be read alongside the matched A_{raw} contrast rather than in place of it. Four randomly initialized Clause networks with the same architecture provide an untrained-weight check. Label permutation provides a separate chance null. The Logit variant lacks a constraint-message channel, so its structurally empty A_{raw} is not used as an informative head-to-head baseline.

Paired inference and interpretation. Table 5 reports the prespecified contrasts $H_{\text{post}} - A_{\text{raw}}$, $H_{\text{post}} - B$, $H_{\text{post}} - (B + A_{\text{raw}})$, and $H_{\text{post}} - H_{\text{pre}}$. We aggregate differences over the selected settings and four independently trained models within a problem; 95% intervals come from 2,000 hierarchical-bootstrap resamples at the model and instance levels. For SAT and NAE-SAT, the state–aggregate differences remain positive after dimension matching (SAT $+.077$ [.064,.091] Macro-F1; NAE-SAT $+.074$ [.069,.079]). Random-weight post-update states are weaker than trained ones by approximately .25–.26 Macro-F1 in SAT and .18–.20 in NAE-SAT across their three densities. Coloring's dimension-matched difference is $-.006$ [$-.016$,$+.004$] R^2, while its trained state beats the random-weight state by .73–.78 R^2, locating accessibility in the trained messages. Max-Cut's dimension-matched difference is $-.027$ [$-.057$,$+.024$] R^2 and its trained–random-weight difference changes sign across settings. At one prespecified representative density per problem, the real H_{post} score exceeds the range of 20 label permutations. Together, these controls establish problem-specific provenance for the accessible statistics.

F EXTENDED DISCUSSION AND LIMITATIONS

The primary result is an evaluation failure hidden by approximation ratio: within the studied OPT-GNN progression, models with similarly high average constraint satisfaction can have sharply different exact-feasibility rates. In SAT and NAE-SAT, stronger feasibility coincides with two further observations: assignment-local stability is more linearly accessible from the variable embeddings, and iterative updates are more strongly aligned with that statistic. Together these observations provide a representation-and-dynamics account of the gap.

The extensions delineate this account across problems. On generic Max-Cut graphs, local-margin accessibility and cut quality vary independently. In graph coloring, OPTGNN-CLAUSE encodes the local margin almost perfectly across densities, including a range with zero exact solves. On structured SATLIB, improved approximation and Hamming distance rarely reach feasibility. These cases establish problem- and distribution-dependent relationships among representation, iterative correction, and exact solving.

Constraint-level aggregation creates a direct alternative interpretation of the probes: the target summarizes incident constraint outcomes, so its decodability may reflect retention of an architecture-adjacent local count. Table 5 resolves this provenance question. For SAT and NAE-SAT, the trained post-update state predicts support more accurately than the exact incoming message aggregate or assignment-conditioned handcrafted statistics, even when the latter are combined; the dimensionality-matched comparison agrees. For planted coloring, the aggregate already matches the state and the combined control exceeds it; for Max-Cut, the comparison is inconclusive. A high probe score therefore has problem-specific provenance, with SAT and NAE-SAT exhibiting learned, linearly accessible organization beyond the tested inputs.

Several directions remain open. OPTGNN also studies problems such as Vertex Cover, which differ from the cut-based or partition-based CSPs considered here. In Vertex Cover, the solution is defined by membership in a selected set rather than by assigning variables to competing partitions, and it is unclear what combinatorial property should play the role of confidence. Identifying useful local-stability statistics in such settings is an important direction for future work. A related question is how to extend this view to pure optimization problems, such as Max-Cut, where there is generally no exact satisfying assignment to serve as a target solution. Appendix J examines an analogous local margin. It is highly accessible in OPTGNN-CLAUSE even when generic-graph cut quality is not improved; on a planted family, the same variant also has the highest near-target attainment rate around the detectability threshold. We report these as two distinct observations rather than treating accessibility as their cause.

Our study also has limitations. First, the structured SATLIB evaluation shows that improvements in approximation and distance rarely reach exact feasibility; the strong solving results are concentrated on random or planted distributions. Second, our analysis is restricted to a family derived from OPTGNN. The crossed SAT ablation isolates a strong interaction between constraint aggregation and the logit objective at $c = 3.5$–3.7, but its new aggregation-plus-SDP cell has three training seeds and does not establish universal component necessity. Third, the cross-instance result replicates over four independently trained models, and the matched-input controls reject direct raw-message retention for SAT/NAE-SAT. The probes establish a shared linearly accessible direction. The SAT flip statistics provide a single-model exploratory view. Future intervention tests can identify the responsible learned operation and account for nonlinear functions of the raw messages. Fourth, all four problems studied here share a partition-like structure: each variable is assigned to one of several competing groups, and the local statistic summarizes how incident constraints support the current choice. We do not know whether an analogous diagnostic exists for asymmetric membership problems such as Vertex Cover. Generalization beyond partition-style CSPs remains open. Fifth, satisfiability certification is auxiliary rather than instance-matched at the $n = 1000$ headline scale. It supports the interpretation below the boundary, but near the random SAT and NAE-SAT thresholds failure can still reflect either an unsatisfiable input or a solver miss.

G SUPPORT AS A SIGNAL

G.1 AN ILLUSTRATIVE SUPPORT SIGNAL UNDER SYMMETRIC CONFIDENCE

Insight 1 (The SAT loss induces a strict ordering of relaxed clause patterns)**.** *Consider the relaxed SAT clause-unsatisfaction loss*

$$u_C^{\mathrm{SAT}}(x) = \prod_{i \in C^+} (1 - x_i) \prod_{i \in C^-} x_i.$$

For an ϵ-relaxed Boolean assignment, where each true literal has false-indicator ϵ and each false literal has false-indicator $1 - \epsilon$, with $0 < \epsilon < 1/2$, the clause patterns satisfy

$$u_C(TTT) < u_C(TTF) < u_C(TFF) < u_C(FFF).$$

Thus, under this symmetric-confidence construction, uniquely satisfied clauses receive the largest relaxed unsatisfaction value among satisfied clauses. The calculation illustrates one route by which support can become distinguishable; it is not a monotonicity claim for arbitrary heterogeneous logits or complete network updates.

Proof. For a positive literal $i \in C^+$, the literal is false when $x_i = 0$, so its relaxed false-indicator is $1 - x_i$. For a negative literal $i \in C^-$, the literal $\neg x_i$ is false when $x_i = 1$, so its relaxed false-indicator is x_i. Therefore,

$$u_C^{\mathrm{SAT}}(x) = \prod_{i \in C^+} (1 - x_i) \prod_{i \in C^-} x_i.$$

Under an ϵ-relaxed Boolean assignment, every true literal has false-indicator ϵ, and every false literal has false-indicator $1 - \epsilon$, regardless of whether the literal is positive or negated. Hence, for a 3-literal clause,

$$u_C(TTT) = \epsilon^3, \qquad u_C(TTF) = \epsilon^2(1 - \epsilon),$$
$$u_C(TFF) = \epsilon(1 - \epsilon)^2, \qquad u_C(FFF) = (1 - \epsilon)^3.$$

Since $0 < \epsilon < 1/2$, we have $\epsilon < 1 - \epsilon$. Therefore,

$$\epsilon^3 < \epsilon^2(1 - \epsilon) < \epsilon(1 - \epsilon)^2 < (1 - \epsilon)^3,$$

and so

$$u_C(TTT) < u_C(TTF) < u_C(TFF) < u_C(FFF).$$

Thus, among satisfied clauses, the uniquely satisfied pattern TFF is closest to violation under the relaxed loss. Aggregated over clauses, variables that uniquely satisfy many clauses repeatedly participate in these high-risk satisfied clauses, providing a local confidence signal.

Moreover, this high-risk status has a stabilizing derivative. In the positive case, $TFF = (1 - \epsilon, \epsilon, \epsilon)$ gives

$$\frac{\partial u_C}{\partial x_T} = -(1 - \epsilon)^2,$$

so gradient descent increases the unique true literal and moves the clause away from the violated FFF pattern. Thus, TFF is not only closest to violation in value; it also produces a direct local push preserving the literal that prevents violation.

□

Insight 2 (The SDP loss mixes support with vector-moment noise)**.** *The SDP-style clause objective does not expose support as a clean pattern-level signal. At any fully rounded assignment its value equals 1 for every satisfying pattern (including TTT, TTF, and TFF), so the rounded patterns are exactly degenerate and carry no support ordering. Away from rounded points its value is determined by continuous inner-product moments of the lifted vector solution, so any support-related signal is mixed with geometry-dependent variation, making it harder for the model to isolate and use.*

Proof. Consider the SDP-style clause contribution

$$\Phi_C(V) = \frac{1}{8}\Bigg(7 - \tau_i\tau_j\tau_k\frac{1}{3}\left[\langle v_i, v_{jk}\rangle + \langle v_j, v_{ik}\rangle + \langle v_k, v_{ij}\rangle\right] - \tau_i\tau_j\frac{1}{2}\left[\langle v_i, v_j\rangle + \langle v_{ij}, v_\emptyset\rangle\right] - \tau_i\tau_k\frac{1}{2}\left[\langle v_i, v_k\rangle + \langle v_{ik}, v_\emptyset\rangle\right] - \tau_j\tau_k\frac{1}{2}\left[\langle v_j, v_k\rangle + \langle v_{jk}, v_\emptyset\rangle\right] - \tau_i\langle v_i, v_\emptyset\rangle - \tau_j\langle v_j, v_\emptyset\rangle - \tau_k\langle v_k, v_\emptyset\rangle\Bigg). \tag{3}$$

Here, $\tau_i, \tau_j, \tau_k \in \{-1, +1\}$ encode the signs of the literals in clause C. The vector v_i is the SDP vector associated with variable x_i, while v_{ij}, v_{ik}, and v_{jk} are lifted SDP vectors associated with the corresponding variable pairs. The vector $v_\emptyset$ is the distinguished reference vector, representing the constant or empty monomial. Since Φ_C is trained by gradient descent to be maximized (the implementation minimizes $-\Phi_C$), the coordinate convention is fixed by requiring this maximization to favor satisfying assignments: literal i is true exactly when $\tau_i\langle v_i, v_\emptyset\rangle = -1$ (equivalently, an unnegated variable is true when its vector is anti-aligned with $v_\emptyset$), which gives $a_i(V) = +1$ when literal i is true and -1 when false, and makes Φ_C larger at fully-satisfied than at violated rounded patterns, as required for training to push toward satisfying assignments rather than away from them. Thus, the terms in Equation (3) are inner-product moments of the lifted SDP representation: unary moments such as $\langle v_i, v_\emptyset\rangle$, pairwise moments such as $\langle v_i, v_j\rangle$ and $\langle v_{ij}, v_\emptyset\rangle$, and higher-order lifted moments such as $\langle v_i, v_{jk}\rangle$.

Define the scalar vector moments

$$a_i(V) = -\tau_i\langle v_i, v_\emptyset\rangle,$$
$$b_{ij}(V) = \tau_i\tau_j\frac{1}{2}\left[\langle v_i, v_j\rangle + \langle v_{ij}, v_\emptyset\rangle\right],$$

and

$$c_{ijk}(V) = -\tau_i\tau_j\tau_k\frac{1}{3}\left[\langle v_i, v_{jk}\rangle + \langle v_j, v_{ik}\rangle + \langle v_k, v_{ij}\rangle\right].$$

Then

$$\Phi_C(V) = \frac{1}{8}\left(7 + a_i(V) + a_j(V) + a_k(V) - b_{ij}(V) - b_{ik}(V) - b_{jk}(V) + c_{ijk}(V)\right).$$

The clause value is therefore a function of scalar inner-product moments of the SDP vectors. These moments are continuous geometric quantities; they are not Boolean truth values and are not determined by the rounded Boolean pattern alone.

To make this explicit, compare two rounded satisfying patterns that share literals i and k, but differ in the middle literal:

$$TFF = (i:T,\ j:F,\ k:F), \qquad TTF = (i:T,\ j':T,\ k:F).$$

In TFF, literal i uniquely satisfies the clause. In TTF, the clause is redundantly satisfied. At a fully rounded V, direct substitution using the coordinate convention above gives $a_i(V) = +1$ when literal i is true and -1 when false; $b_{ij}(V) = +1$ when literals i, j agree in truth value and -1 when they disagree; and $c_{ijk}(V)$ equals the product of the three literals' truth-signs (+1 for true, -1 for false), i.e., $+1$ under an even number of false literals and -1 under an odd number. Below, a_i, b_{ij}, c_{ijk} (no argument) denote these values folded into the generic formula's own coefficient for each term – $+1$ for the a and c terms and -1 for the b terms, matching $\Phi_C(V) = \frac{1}{8}(7 + a_i(V) + a_j(V) + a_k(V) - b_{ij}(V) - b_{ik}(V) - b_{jk}(V) + c_{ijk}(V))$ above – so, for instance, an agreeing pair ($b_{ij}(V) = +1$) contributes $-b_{ij}(V) = -1$ and is printed with a minus sign below, while a disagreeing pair ($b_{ij}(V) = -1$) contributes $+1$ and is printed with a plus sign; a and c terms carry no such flip and are printed with their own sign directly. This is exactly the generic formula evaluated at V, not an independently chosen sign, and reproduces it numerically term for term. Using this notation, the corresponding SDP contributions are

$$\Phi_C(TFF; V) = \frac{1}{8}\left(7 + a_i - a_j - a_k + b_{ij} + b_{ik} - b_{jk} + c_{ijk}\right),$$

and

$$\Phi_{C'}(TTF; V) = \frac{1}{8}\left(7 + a_i + a_{j'} - a_k - b_{ij'} + b_{ik} + b_{j'k} - c_{ij'k}\right).$$

Subtracting gives

$$\Phi_C(TFF;V) - \Phi_{C'}(TTF;V) = \frac{1}{8}\left(-a_j - a_{j'} + b_{ij} + b_{ij'} - b_{jk} - b_{j'k} + c_{ijk} + c_{ij'k}\right).$$

The shared terms a_i, a_k, and b_{ik} cancel. The remaining terms depend on the non-shared literals j and j', and on their SDP moments with i and k. These moments need not be equal, and their comparison is not fixed by the rounded support pattern.

Evaluated at a fully rounded V, the difference is exactly zero: substituting the truth-signs $s \in \{\pm 1\}$ gives $\Phi_C = \frac{1}{8}\big(7 + \sum s - \sum s_i s_j + s_i s_j s_k\big)$, which equals 1 for TTT, TTF, and TFF and 0 for FFF. The uniquely supported pattern TFF and the redundantly satisfied pattern TTF are therefore indistinguishable at rounded points, whereas the logit loss separates them strictly. At relaxed (non-rounded) configurations the sign of the difference depends on the continuous moments and no universal ordering holds.

The consistency and unit-norm penalties used in Yau et al. (2024) restrict the feasible vector geometry, but they do not make the SDP clause value a function only of the rounded Boolean pattern. Even under strong consistency, the value remains determined by continuous moment-like quantities of the vector solution.

Hence the SDP loss may contain information correlated with support, but it does not expose support as a clean local signal in the same way as the logit loss. The support-related effect is mixed with geometry-dependent variation in the vector moments, which acts as noise from the perspective of support detection and may make the signal harder to isolate linearly. □

H NAE TABLES

This appendix reports the NAE-SAT tables referenced from Section 4: within-instance (5-fold cross-validated) linear accessibility of support by density (Table 7; the cross-instance transfer results are in the transfer table below) and the probe-matched four-model regime sweep (Table 8).

At the sparsest density tested ($c = 0.5$), OPTGNN and OPTGNN-IR solve almost nothing while OPTGNN-CLAUSE solves every instance; the SAT low-density audit in Appendix C identifies decoding budget as the source of this low rate. Re-decoding the identical checkpoints' identical forward passes with $\mathrm{RH_ROUNDS} = 1000$ instead of the matched-budget 10 lifts OPTGNN-IR to 58–100% solved and OPTGNN to 46–90% solved by $c = 1.0$, with no retraining.

I SATLIB EVALUATION

SATLIB (Hoos & Stützle, 2000) is a standard public repository of SAT benchmark instances used to evaluate solvers beyond synthetically generated data. We evaluate on 13 structured, non-random, largely real-world-derived benchmark families (Appendix I.1), deliberately excluding SATLIB's random benchmarks: our own synthetic training and evaluation data (Section 4) already cover the uniform-random regime, so the external benchmark's value is specifically in testing structured, non-random, real-world-derived instances that our own data generation cannot produce.

I.1 STRUCTURED, NON-RANDOM BENCHMARK FAMILIES

We evaluate on 13 SATLIB families that are structured rather than uniform-random, several of them SAT encodings of genuine real-world problem instances rather than synthetic constructions: *Flat* and *Morphed graph colouring* (structured, generated); *Blocksworld* and *Logistics* (SAT encodings of AI planning-competition domain instances); *SSA* (real single-stuck-at circuit fault diagnosis instances); *Quasigroup* completion; *Parity* learning; *Inductive inference* (*II*); *Towers of Hanoi*; and *DUBOIS*, *BF*, and *Pigeonhole*, three families that are unsatisfiable by construction. We also attempted *BMC* (bounded model checking on real IBM hardware-verification circuits), but every instance exceeded our compute budget after reduction (below) and none could be evaluated.

None of these families are natively 3-SAT: clause widths range from unit clauses up to several dozen literals. Since the SDP embedding-update gradient shared by all four variants (Section 3) is defined only for exactly-3-literal clauses, we first reduce each instance to an equisatisfiable 3-CNF formula.

Table 7: Linear accessibility of NAE-SAT support. For each $n = 1000$ NAE-SAT instance and each valid adjacent support pair $(0, 1)$, $(1, 2)$, and $(2, 3)$, we train a linear SVM using 5-fold cross-validation in the learned representation space. Each cell reports Macro-F1 $\pm$ standard deviation, where Macro-F1 is first averaged over folds and then over instances, and the standard deviation is computed across instances. Higher values indicate clearer geometric organization by support.

c	Model	$(0, 1)$	$(1, 2)$	$(2, 3)$
0.5	OPTGNN-IR-NAE	.46±.02	.44±.04	–
	OPTGNN-LOGIT-NAE	.73±.04	.59±.05	.56±.09
	OPTGNN-CLAUSE-NAE	**.91±.03**	.76±.06	.55±.10
1.0	OPTGNN-IR-NAE	.45±.02	.47±.04	.48±.07
	OPTGNN-LOGIT-NAE	.74±.02	.59±.04	.54±.06
	OPTGNN-CLAUSE-NAE	**.92±.02**	**.82±.03**	.61±.07
1.5	OPTGNN-IR-NAE	.48±.02	.45±.04	.46±.06
	OPTGNN-LOGIT-NAE	.70±.02	.59±.03	.53±.06
	OPTGNN-CLAUSE-NAE	**.95±.01**	**.87±.02**	**.70±.06**
1.8	OPTGNN-IR-NAE	.48±.02	.46±.04	.44±.07
	OPTGNN-LOGIT-NAE	.67±.02	.59±.03	.53±.05
	OPTGNN-CLAUSE-NAE	**.94±.02**	**.87±.03**	**.72±.06**
2.0	OPTGNN-IR-NAE	.49±.02	.47±.03	.46±.06
	OPTGNN-LOGIT-NAE	.64±.02	.59±.03	.54±.05
	OPTGNN-CLAUSE-NAE	**.93±.01**	**.86±.02**	**.73±.04**
2.1	OPTGNN-IR-NAE	.49±.02	.48±.02	.48±.05
	OPTGNN-LOGIT-NAE	.63±.02	.60±.03	.55±.05
	OPTGNN-CLAUSE-NAE	**.93±.01**	**.85±.02**	**.73±.04**
2.5	OPTGNN-IR-NAE	.51±.02	.48±.03	.47±.04
	OPTGNN-LOGIT-NAE	.59±.02	.58±.03	.54±.04
	OPTGNN-CLAUSE-NAE	**.91±.01**	**.84±.02**	**.73±.04**

Table 8: Four-model NAE-SAT regime sweep for $n = 1000$, 400 instances per density and variant (100 instances $\times$ 4 independently trained models), underlying the NAE-SAT discussion in Section 4. Intervals are computed at the model level (mean $\pm$ t-based 95% CI across the four model-level rates, $\mathrm{df} = 3$), not pooled across the 400 instances, since the four trained models are the independent unit. OPTGNN-CLAUSE's exact-solve rate collapses sharply between $c = 1.5$ (94.8%, 95% CI $[89.2, 100]$) and $c = 1.75$ (12.5%, CI $[3.0, 22.0]$), while its held-out cross-instance probe Macro-F1 stays within .769–.817 across the entire 0.5–2.1 range.

c	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE solved	CLAUSE probe F1
0.50	.00	.00	.00	**1.000** [1.00,1.00]	.788 [.73,.85]
1.00	.01	.04	.38	**1.000** [1.00,1.00]	.801 [.75,.85]
1.25	.00	.00	.05	**.998** [.99,1.00]	.812 [.77,.86]
1.50	.00	.00	.07	**.948** [.89,1.00]	.817 [.77,.87]
1.75	.00	.00	.07	**.125** [.03,.22]	.769 [.74,.81]
1.80	.00	.00	.07	**.033** [.00,.08]	.772 [.74,.81]
1.90	.00	.00	.00	.000 [.00,.00]	.782 [.75,.82]
2.00	.00	.00	.00	.000 [.00,.00]	.795 [.76,.84]
2.10	.00	.00	.00	.000 [.00,.00]	.791 [.75,.83]

For clauses of width $k > 3$ this reduction chains in $k - 2$ auxiliary variables, which lengthens the constraint graph's effective diameter relative to the original instance; because message-passing GNNs are known to struggle with long-range dependencies, this reduction is a plausible additional contributor to the near-zero solving rates below, alongside these families' structured, non-random origin. As a partial check, we correlated each instance's reduction-expansion factor (reduced size over original size) against its approximation ratio: pooled across all 296 instances this correlation is confounded by family membership, since several families (e.g., *Flat/Morphed graph colouring*, *DUBOIS*) have essentially uniform expansion within the family, so any pooled trend partly reflects between-family differences rather than the reduction itself. Restricted to families with enough within-family variance to test (*ais*, *ii*, *parity*, *pigeonhole*, *bf*, *blocksworld*, *ssa*), OPTGNN-CLAUSE's within-family correlations are mixed in sign and mostly not significant, and where significant (*ais*, *parity*) are positive rather than negative – the opposite of what the topological-confound hypothesis predicts. This does not rule out the confound, since most families offer little within-family contrast to test against and approximation ratio is a coarser proxy than solve rate (which has almost no variance to correlate against here), but it does not provide positive evidence for it either. The reduction proceeds as follows: literals are deduplicated within each clause and tautological clauses (containing both a literal and its negation) are dropped; clauses of width 1 or 2 are padded to width 3 using fresh auxiliary variables with compensating clauses that preserve full logical equivalence (not merely equisatisfiability); and clauses of width $k > 3$ are split into $k - 2$ three-literal clauses via a standard auxiliary-variable chain, which preserves satisfiability and lets any solution to the reduced formula, restricted to the original variables, be extended to satisfy the original clause. This reduction was verified against hand-constructed cases (including unit, dedup, tautological, forced-UNSAT, and mixed-width instances) by exhaustively enumerating satisfying assignments of both the original and reduced formulas with CP-SAT and checking exact equisatisfiability and solution-extendability. The model only ever sees the reduced instance; all reported metrics (solved status, approximation ratio, Hamming distance) are computed by projecting the decoded assignment back onto the *original* variables and scoring against the *original*, unreduced clauses, so they reflect the real problem rather than reduction artifacts. Instances whose reduced size exceeds a fixed compute budget are excluded (columns below report n, the number of instances retained per family).

Table 9: Mean approximation ratio by SATLIB family (structured, non-random). n is the number of instances of that family whose 3-CNF reduction fit the compute budget; the same n instances are scored for every variant.

Family	n	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
All-interval-series	4	**0.991**	0.926	0.990	0.990
BF (UNSAT)	4	0.959	0.966	0.991	**0.992**
Blocksworld	4	0.945	0.979	**0.991**	0.990
DUBOIS (UNSAT)	13	0.918	0.905	**0.995**	0.995
Flat graph colouring	100	0.922	0.918	0.917	**0.990**
Hanoi	1	0.984	0.964	**0.990**	0.989
Inductive inference	23	0.946	0.957	0.976	**0.982**
Logistics	3	0.975	0.950	0.988	**0.993**
Morphed graph colouring	100	0.968	**0.973**	0.968	0.970
Parity	30	0.973	0.914	0.984	**0.992**
Pigeonhole (UNSAT)	5	0.953	0.952	0.968	**0.975**
Quasigroup	2	0.871	0.918	0.980	**0.981**
SSA	7	0.971	0.975	**0.993**	0.991

Solving is essentially absent on these harder, structured instances: the only exact solve anywhere in Tables 9–10 is OPTGNN-LOGIT solving 1 of 4 BLOCKSWORLD instances; every other variant–family combination solves zero. The improvement from OPTGNN to OPTGNN-CLAUSE is therefore in *closeness* rather than solving capacity. Pooled paired tests favor OPTGNN-CLAUSE, but the two largest families contribute 200 of the 296 approximation-ratio pairs; we therefore foreground the family-level direction. OPTGNN-CLAUSE has a better mean than OPTGNN in 12 of 13 families on approximation ratio (two-sided family sign test, $p = .0034$) and all 9 families on Hamming distance ($p = .0039$). Across the cumulative hierarchy, the relaxed-label readout carries most of the approximation gain: OPTGNN-LOGIT exceeds plain OPTGNN on 10 of 13 families. Constraint aggregation adds a smaller, family-dependent increment: OPTGNN-CLAUSE exceeds OPTGNN-

Table 10: Mean normalized Hamming distance to the nearest satisfying assignment (upper bound: CP-SAT is not always run to proved optimality within budget), unsolved instances only, for the families where CP-SAT returned a candidate nearest solution within the time budget for every variant (excludes the UNSAT-by-construction families, where the metric is undefined, and BMC/Inductive inference, where CP-SAT returned none within budget). Lower is better.

Family	n	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
All-interval-series	4	0.134	0.238	**0.126**	0.126
Blocksworld	3	0.194	0.220	**0.154**	0.184
Flat graph colouring	100	0.324	0.363	0.324	**0.312**
Hanoi	1	0.214	0.265	0.242	**0.188**
Logistics	3	0.180	0.250	**0.103**	0.108
Morphed graph colouring	100	0.199	**0.170**	0.196	0.185
Parity	20	0.378	0.415	0.459	**0.314**
Quasigroup	1	0.385	0.230	0.125	**0.119**
SSA	4	0.224	0.247	0.267	**0.208**

LOGIT on 8 families, trails it on 3, and ties on 2. Recurrence alone contributes no systematic gain: OPTGNN-IR exceeds plain OPTGNN on 6 of 13 families. (Two families are CNF encodings of graph-coloring instances; Appendix K instead evaluates native q-ary coloring.)

J EXTENSION TO MAX-CUT

We use Max-Cut as a boundary test of the SAT account. We specialize the logit-space framework to $k = 2$ and separately ask whether an assignment-local edge margin is linearly accessible and whether the variants approach a planted objective target. Generic graphs decouple accessibility from cut quality; the planted family supplies a target-attainment test.

Architecture and loss. OPTGNN and OPTGNN-IR's SDP-relaxation objective L (the abstract "SDP-based objective" of Section 3's base model) specializes at $k = 2$ to the Goemans–Williamson Max-Cut SDP objective, per edge (i, j) with embeddings v_i, v_j on the unit sphere:

$$L^{\mathrm{SDP}}_{\mathrm{edge}}(v_i, v_j) = \tfrac{1}{2}\big(1 + \langle v_i, v_j\rangle\big), \tag{4}$$

0 when v_i, v_j are antipodal (cut), 1 when identical (not cut), summed over edges and used for both the gradient driving the shared update recurrence and, for OPTGNN/OPTGNN-IR, the training loss itself. For an edge (i, j) with relaxed truth values $x_i, x_j \in [0, 1]$, the Max-Cut relaxed-label loss is

$$u^{\mathrm{MaxCut}}_C(x_i, x_j) = x_i x_j + (1 - x_i)(1 - x_j), \quad C = (i, j) \in E, \tag{5}$$

the probability both endpoints land on the same side, summed over edges, with $x_i = \sigma(\ell_i)$ for a raw readout logit ℓ_i. Unlike SAT's clauses, whose literal signs already break the symmetry of a uniform-confidence start, Max-Cut's edges are unsigned: $\ell_i = 0\,\forall i$ (every node at $x_i = \frac{1}{2}$) is an exact zero-gradient fixed point of Eq. (5), so nothing in the raw loss favors moving away from it. During training only, we add a bounded anti-collapse regularizer, $-\lambda \tanh(\ell_i)^2$ averaged over nodes ($\lambda = 0.05$), zero exactly at this fixed point and strictly negative everywhere else, turning it from a stable attractor into an unstable one (a term that only decreases the loss away from $\ell_i = 0$ makes that point a local loss maximum rather than minimum along this direction, so gradient descent moves away from it instead of toward it); the $\tanh$ bound prevents it from being gamed by unboundedly inflating $|\ell_i|$ regardless of whether the resulting prediction is correct. The SDP-style OPTGNN/OPTGNN-IR baselines receive no equivalent training-time term, since their symmetric fixed point is not degenerate in the same way. We ablated this bonus for OPTGNN-LOGIT on Max-Cut, replicated across training seeds (half the full training budget): with the bonus, training loss leaves the fixed-point value after about 150 steps; without it, only after about 600 steps, a fourfold delay. At matched step count the bonus condition finishes ahead on every metric we measured – lower training loss (.431 vs. .475), higher cut ratio (.795 vs. .787), and larger mean $|\ell_i|$ (1.91 vs. 1.84) – consistent with the bonus doing genuine, quantifiable work: shortening time spent at the degenerate point and leaving the model further along after the same amount of training. The zero-gradient point is a fixed point of the idealized objective, not an absorbing state of real training with its other sources of asymmetry (e.g., each node's

distinct local graph neighborhood) – the unboosted run does eventually leave it, just slower and behind. Because the unboosted run still reaches cut ratios in the same range the SDP-style baselines already achieve on generic graphs without any such term (Table 11), the bonus does not appear to manufacture an advantage over those untreated baselines for the logit-space variants; its measured effect is a training-efficiency one internal to this readout, not the source of the cross-architecture comparisons reported in Section 4, which turn on confidence-accessibility and target-attainment rather than raw cut ratio. This ablation also targets the generic (non-planted) Max-Cut setting, where, as established above, no achievable near-perfect cut exists and there is consequently no discrete solving event for escape speed to gate; it therefore speaks to the bonus's role in approximation-quality dynamics, not to whether faster escape matters for reaching a discrete target, which is only tested on the planted family (Table 12) and was not separately ablated. OPTGNN-CLAUSE's constraint-level aggregation specializes the same way: the clause representation $c_j = \mathrm{Norm}(M'(\sum_{v_i \in C_j} v_i))$ of Section 3 becomes, for an edge (i, j), an edge representation

$$c_{ij} = \mathrm{Norm}(M'(v_i + v_j)), \tag{6}$$

fed back into each endpoint's update exactly as before, $v_{i+1} = \mathrm{Norm}(M(v_i, \nabla_i, \sum_{j:(i,j)\in E} c_{ij}))$: constraints of arity 2 (edges) replace constraints of arity 3 (clauses), with no other change to the mechanism. OPTGNN-LOGIT and OPTGNN-IR specialize analogously – the sigmoid readout and the shared recurrent update are both arity-agnostic and require no modification at $k = 2$. We define an analogous confidence signal for a node v – the same operational g_ϕ notion of a learned concept from Section 2, specialized here to Max-Cut's arity-2 constraints in place of SAT's support – as

$$\mathrm{confidence}(v) := \#\{\text{edges of } v \text{ crossing the cut}\} - \#\{\text{edges of } v \text{ staying on } v\text{'s side}\}, \tag{7}$$

the direct $k = 2$ specialization of support (Section 2): a large magnitude means v's neighbors mostly agree on which side v belongs to, near zero means they are split.

Data. We evaluate on eight datasets. Four are the synthetic random graph families used in Yau et al. (2024)'s own Max-Cut experiments, 400–500 nodes per graph: Barabási–Albert ($m = 4$), Erdős–Rényi ($p = 0.15$), Holme–Kim ($m = 4$, $p = 0.25$), and Watts–Strogatz ($k = 4$, $p = 0.25$); following Yau et al. (2024)'s methodology, a fresh graph is sampled from the generator at every training step (never a pre-materialized pool), for 4,000 steps, with 100 held-out instances per dataset for evaluation. Four are real-world graphs from the TU benchmark suite (Morris et al., 2020), analogous in role to SATLIB above: ENZYMES, COLLAB, and REDDIT-MULTI-12K/REDDIT-MULTI-5K, each with a random 100-graph held-out test split (of which 100 are used for evaluation here) and training on the remainder for the same 4,000 steps. All four controlled variants are trained from scratch on each of the eight datasets.

Table 11: Local-margin decodability (R^2 of a linear probe) | mean cut ratio, across all four controlled variants and all eight Max-Cut datasets: four synthetic random graph families following Yau et al. (2024), and four real-world TU graphs (Morris et al., 2020). OPTGNN-CLAUSE's final state makes the margin more linearly accessible than the other variants' states on all eight datasets; raw cut ratio itself remains close across variants. The selected planted Max-Cut raw-message control is inconclusive, so this figure does not isolate where the accessible signal is formed.

Dataset	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
Barabási–Albert	.00|.72	.01|.72	.09|.71	**.60**|.71
Erdős–Rényi	.02|.57	.01|.57	.77|.58	**.84**|.58
Holme–Kim	.00|.70	.02|.70	.15|.70	**.63**|.69
Watts–Strogatz	.01|.79	.00|.79	.44|.79	**.73**|.79
ENZYMES	−.01|.78	−.02|.77	.31|.77	**.48**|.77
COLLAB	.03|.59	.07|.59	.14|.56	**.47**|.58
REDDIT-M-12K	.00|.94	.00|.93	.02|.94	**.45**|.93
REDDIT-M-5K	.00|.95	.00|.95	.00|.95	**.51**|.94

OPTGNN and OPTGNN-IR carry essentially no linearly accessible local-stability signal on any of the eight datasets ($R^2 \leq 0.07$ throughout), OPTGNN-LOGIT is intermediate and dataset-dependent (R^2 from ≈ 0 to 0.77), and OPTGNN-CLAUSE is the strongest on every dataset ($R^2 = 0.45$–0.84),

qualitatively mirroring the SAT transfer ordering in Figure 1(c–d). Cut ratio is within .03 across variants on every dataset except COLLAB; no variant is consistently best or worst. This decoupling is informative: none of these datasets has an achievable near-perfect cut, so there is no last-mile feasibility event for an accessible local-stability signal to accompany.

A genuine approximation-to-solving transition (planted model). To construct a Max-Cut instance family that does have such a gap, we use a planted q-community partition (stochastic block model) – the same general model graph coloring specializes to $q = 3$ in Appendix K. Given n nodes split into q equal groups, we place an edge between nodes in *different* groups with probability p_{out} (the recoverable structure) and between nodes in the *same* group with probability p_{in} (noise); Max-Cut is the $q = 2$ case studied in this section. We parametrize instances by expected degree rather than raw probability,

$$b := \tfrac{n}{2}\, p_{\text{out}}, \qquad a := \tfrac{n}{2}\, p_{\text{in}},$$

holding b fixed and sweeping a: detectability of the planted partition is governed by expected degree, not raw probability. Under this parametrization, a and b are the within- and cross-block contributions to expected degree. Converting the standard symmetric two-block Kesten–Stigum condition to these units gives (Decelle et al., 2011)

$$(a - b)^2 = a + b, \tag{8}$$

with weak detection possible on the disassortative side when $(a - b)^2 > a + b$. For $b = 12$, the relevant boundary is $a \approx 7.58$. The reported sweep $a \in \{0, 2, 4, 6\}$ approaches but does not cross this boundary; it tests degradation toward the detectability limit rather than an information-theoretically impossible regime.

We evaluate all four variants at $a \in \{0, 2, 4, 6\}$, $b = 12$ fixed, at three sizes $n \in \{200, 400, 800\}$, with 12 held-out instances per (n, a) configuration, using checkpoints trained on generic Erdős–Rényi-style instances (never on planted graphs). For each (n, a) we report the planted-partition cut ratio—a known achievable reference—and, per variant, NEARTARGET | mean achieved cut ratio. NEARTARGET is the fraction with achieved ratio at least the planted ratio minus 0.005; solutions that improve on the planted partition count as attaining the target. Partition recovery and exact Max-Cut optimality are separate metrics.

Table 12: All four variants across the planted-partition hardness sweep ($b = 12$ fixed), NEARTARGET fraction | cut ratio. NEARTARGET means achieved cut ratio $\geq$ planted reference-0.005; partition recovery and exact optimality are separate metrics. At $a = 4$, OPTGNN-CLAUSE's target-attainment fraction is 1.7–3.3$\times$ that of the next-best variant at every size tested.

n	a	Ideal	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
200	0	1.000	1.00|1.00	1.00|1.00	.00|.64	1.00|1.00
200	2	0.860	1.00|.86	1.00|.86	.00|.55	1.00|.86
200	4	0.754	.42|.71	.33|.71	.00|.54	**1.00|.75**
200	6	0.668	.00|.60	.00|.60	.00|.51	.00|.64
400	0	1.000	1.00|1.00	1.00|1.00	.00|.65	1.00|1.00
400	2	0.857	1.00|.86	1.00|.86	.00|.54	1.00|.86
400	4	0.750	.25|.69	.25|.72	.00|.53	**.83|.75**
400	6	0.665	.00|.60	.00|.61	.00|.52	.00|.64
800	0	1.000	1.00|1.00	1.00|1.00	.00|.57	1.00|1.00
800	2	0.857	1.00|.86	1.00|.86	.00|.54	1.00|.86
800	4	0.750	.33|.69	.25|.72	.00|.52	**.58|.75**
800	6	0.667	.00|.62	.00|.60	.00|.52	.00|.64

At $a = 4$, the mean cut-ratio difference is modest, but OPTGNN-CLAUSE reaches the planted objective tolerance on 0.58–1.00 of instances across the three sizes, versus 0.25–0.42 for OPTGNN, 0.25–0.33 for OPTGNN-IR, and 0 for OPTGNN-LOGIT. Its mean cut ratio also tracks the planted target closely at every size. Thus, local-margin accessibility and stronger target attainment coincide in this planted regime, although the generic datasets above show that the two observations decouple elsewhere. OPTGNN-LOGIT's zero target attainment, even on the bipartite $a = 0$ instances, is a fixed-point effect rather than a broken arm: its trained logits stay near the zero-gradient uniform

state (mean |logit| = .19 on $a = 0$ instances at $n = 200$, versus 3.4 for OPTGNN-CLAUSE), so its raw decode is close to chance, whereas aggregation moves the state away from the symmetric fixed point. The anti-collapse regularizer is not what produces OPTGNN-CLAUSE's advantage: retraining it with $\lambda = 0$ (same seed and recipe) leaves both metrics essentially unchanged. On generic Max-Cut ($n = 1000$) the local-margin probe gives $R^2 = .805$ and .746 at $|E|/n = 2$ and 4 (versus .819 and .755 with $\lambda = .05$) at identical mean cut ratios (.825, .729 versus .824, .730), and on planted coloring $R^2 = .968$, .948, and .960 at $c = 2$, 3, and 4.69 (versus .956, .945, and .953).

K EXTENSION TO GRAPH COLORING

Section 3's logit-space framework is stated for a general alphabet size q (Section 2), not just the Boolean case. We specialize it here to $q = 3$ graph coloring, a Max-2-CSP over alphabet $[3]$ in which each edge is an arity-2 "not-equal" constraint, and test the evaluated solver configurations on both solving/approximation and linear accessibility of the current-color agreement margin.

Architecture and loss. Each node is embedded as a single unit vector v_i (one point on the sphere per node, not q separate ones – the direct analog of Max-Cut's v_i), and the SDP relaxation objective driving the shared update recurrence (Section 3; identical for all four variants, only the readout/loss differs) is unchanged in form from Eq. (4)'s Max-Cut case, $L^{\mathrm{SDP}}_{\mathrm{edge}}(v_i, v_j) = \frac{1}{2}(1 + \langle v_i, v_j\rangle)$ per edge: the SDP-style coloring variants deliberately reuse this Max-Cut relaxation; only the decoder depends on q, where the single random hyperplane of the $q = 2$ case is replaced by q reference directions forming a regular $(q-1)$-simplex (pairwise inner product exactly $-1/(q-1)$) in a random 2D subspace; at $q = 2$ this reduces exactly to Max-Cut's decode. OPTGNN-LOGIT's readout becomes a softmax over q colors, $p_i \in \Delta^{q-1}$, whose c-th entry $p_i[c]$ is the relaxed probability that node i takes color c (so $\sum_c p_i[c] = 1$, the softmax analog of SAT's single Boolean logit x_i). For an edge (i, j), $\sum_c p_i[c]\, p_j[c]$ is then the probability i and j are assigned the *same* color – exactly what the "not-equal" constraint forbids – giving edge loss

$$u_C^{\mathrm{Coloring}}(p_i, p_j) = \sum_{c=1}^{q} p_i[c]\, p_j[c], \quad C = (i, j) \in E, \tag{9}$$

the direct q-ary generalization of Eq. (5)'s $q = 2$ same-side probability. As with Max-Cut, the fully uniform state $p_i = (\frac{1}{q}, \ldots, \frac{1}{q})\ \forall i$ is a zero-gradient fixed point of Eq. (9), and edges again have no sign pattern to break the symmetry of a uniform start. During training only, we add an analogous bounded anti-collapse regularizer in probability space rather than logit space, $-\lambda \tanh\Big(\max_c p_i[c] - \frac{1}{q}\Big)^2$ averaged over nodes ($\lambda = 0.05$): zero exactly at the uniform fixed point, strictly negative once any color's probability rises above the uniform baseline $\frac{1}{q}$ (destabilizing it the same way as Max-Cut's bonus above), and $\tanh$-bounded for the same gaming-resistance reason as Max-Cut's. OPTGNN-CLAUSE's edge-level aggregation is unchanged in form from Eq. (6): for an edge (i, j), the edge representation

$$c_{ij} = \mathrm{Norm}(M'(v_i + v_j))\,, \tag{10}$$

fed back into each endpoint's update exactly as in Section 3 and Eq. (6), is identical because the mechanism is alphabet-agnostic and depends only on constraint arity, which remains 2 regardless of q. We define confidence analogously to Max-Cut's – again the same g_ϕ notion of a learned concept (Section 2), now specialized to q-ary alphabets –,

$$\begin{aligned}\mathrm{confidence}(v) := {} & \#\{\text{neighbors of } v \text{ with a different color}\} \\ & - \#\{\text{neighbors of } v \text{ with the same color}\}.\end{aligned} \tag{11}$$

computed from each model's own decoded coloring (which may contain conflicts), not ground truth.

Data. We use the planted proper-coloring model of Alon & Kahale (1994): the $q = 3$, $p_{\mathrm{in}} = 0$ case of the general planted q-community model introduced in Appendix J (with $p_{\mathrm{out}} = p$): partition n nodes into 3 equal classes and place an edge between two nodes, independently, with probability p if and only if they lie in *different* classes. This guarantees exact 3-colorability by construction: with $p_{\mathrm{in}} = 0$, no within-class edges occur. Max-Cut instead increases within-block noise while holding cross-block signal fixed; its reported sweep approaches, but does not cross, the detectability boundary

in Eq. (8). We parametrize density as $c := |E|/n$; since only the $\binom{n}{2} \cdot \frac{2}{3}$ cross-class pairs can receive an edge, $c \approx pn/3$, so $p \approx 3c/n$. Training samples a fresh instance every step for $4{,}000$ steps, with n drawn uniformly from $[100, 300]$ and c drawn uniformly from $[2, 6]$. For this planted family, the relevant structural reference is the planted three-community Kesten–Stigum scale, which corresponds to average degree $d = 4$, or $c = |E|/n = 2$, under our convention. This is a detectability reference rather than a satisfiability threshold: every generated graph remains exactly 3-colorable. Evaluation uses held-out planted instances at $n = 1{,}000$ (never seen at training size), 10 instances per density, across $c \in [1, 20]$ (112 density points: a 0.5 grid refined to 0.02-steps over $[3.5, 5.0]$, plus the added reference point $c = 4.69$), with no retraining – the same train-small/eval-large-and-varied protocol used throughout this paper. Unlike Appendix J's eight Max-Cut datasets, we evaluate graph coloring only on this planted family; we did not construct or evaluate a real-world/generic graph-coloring benchmark analogous to the TU graphs used for Max-Cut, and note this as a scope limitation rather than extrapolate beyond it.

Table 13: Graph coloring solving/approximation: SOLVED fraction (exact valid 3-coloring) | fraction of properly-colored edges, representative densities $c = |E|/n$, $n = 1{,}000$, 10 held-out planted instances per density, no retraining. Full 112-point results in the released code. OPTGNN-CLAUSE is the only variant that ever reaches SOLVED$= 1$ (from $c \approx 4$ on, held through $c = 20$; $83/112$ density points solved exactly, $110/112$ with at least one instance solved). OPTGNN never reaches SOLVED$= 1$ at any of the 112 densities tested and has the lowest mean edge ratio of the four (0.912); OPTGNN-LOGIT solves even less often than plain OPTGNN ($3/112$ density points with any solve at all, versus OPTGNN's $13/112$ and OPTGNN-IR's $19/112$) despite a comparable edge ratio to OPTGNN-IR. Thus solving capacity and approximation quality dissociate across the hierarchy.

c	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
1.0	.00|.989	.00|.991	.20|.999	**1.00|1.000**
1.5	.00|.965	.00|.975	.00|.989	**.80|1.000**
2.0	.00|.948	.00|.944	.00|.968	.00|**.996**
2.5	.00|.924	.00|.929	.00|.945	.00|**.986**
3.0	.00|.909	.00|.911	.00|.923	**.50|.997**
3.5	.00|.900	.00|.891	.00|.929	**.70|1.000**
4.0	.00|.898	.00|.902	.00|.911	**1.00|1.000**
4.69	.00|.890	.00|.935	.00|.936	**1.00|1.000**
6.0	.00|.917	.00|.957	.00|.949	**1.00|1.000**
10.0	.00|.966	.00|.968	.00|.965	**1.00|1.000**
15.0	.00|.928	.30|.938	.00|.943	**1.00|1.000**
20.0	.10|.935	.30|.977	.20|.927	**1.00|1.000**

Across all densities, OPTGNN-CLAUSE has the highest edge ratio and, wherever any variant solves at all, the highest SOLVED fraction. Solving is not monotonic in density for OPTGNN-CLAUSE: it solves perfectly at the sparsest point tested ($c = 1$), dips to SOLVED $= 0$ at $c = 2$–2.5 despite an edge ratio still $\geq .986$, then recovers and holds SOLVED $= 1$ continuously from $c \approx 4$ through $c = 20$. The dip occurs near the planted-family detectability scale $c \approx 2$, even though every instance remains 3-colorable by construction. It therefore marks a difficult inference regime rather than a satisfiability transition. Nor is the ordering among the remaining three variants monotonic in architectural complexity: OPTGNN-LOGIT – which adds a learned logit readout on top of OPTGNN/OPTGNN-IR's shared SDP-relaxation recurrence – is the *worst* solver of the four by density-points-with-any-solve, despite an edge ratio in line with OPTGNN-IR's. Only once edge-level aggregation is added on top of the logit readout (OPTGNN-CLAUSE) does solving capacity follow expectation; the logit readout alone is a liability for solving here, consistent with this paper's broader finding that approximation quality and solving capacity are dissociated quantities.

Size shift. Table 13 above uses a single evaluation size, $n = 1{,}000$; we additionally test whether OPTGNN-CLAUSE's advantage holds as n varies. Using the same checkpoints (trained at $n \in [100, 300]$), we evaluate at $n \in \{200, 500, 1{,}000, 2{,}000\}$ and three representative densities: $c \in \{2.5, 4.69, 10\}$, with 12 held-out instances per (n, c) configuration.

Table 14: Graph coloring size-shift evaluation: SOLVED fraction | edge ratio across $n \in \{200, 500, 1{,}000, 2{,}000\}$ at three representative densities, with 12 held-out instances per configuration. The same checkpoints as Table 13 are used; $n = 200$ lies within the training-size range, whereas $n \in \{500, 1{,}000, 2{,}000\}$ are out of distribution in size.

n	c	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
200	2.5	.00|.931	.00|.941	.00|.957	**.08**|**.989**
200	4.69	.00|.943	.00|.948	.00|.930	**1.00**|**1.000**
200	10.0	.08|.978	.25|.956	.00|.956	**1.00**|**1.000**
500	2.5	.00|.926	.00|.938	.00|.949	**.08**|**.991**
500	4.69	.00|.909	.00|.907	.00|.943	**1.00**|**1.000**
500	10.0	.00|.971	.17|.976	.00|.973	**1.00**|**1.000**
1,000	2.5	.00|.924	.00|.925	.00|.946	.00|**.987**
1,000	4.69	.00|.918	.00|.901	.00|.936	**1.00**|**1.000**
1,000	10.0	.00|.945	.08|.937	.00|.979	**1.00**|**1.000**
2,000	2.5	.00|.922	.00|.922	.00|.940	.00|**.986**
2,000	4.69	.00|.899	.00|.947	.00|.913	**1.00**|**1.000**
2,000	10.0	.08|.962	.17|.980	.00|.953	**1.00**|**1.000**

OPTGNN-CLAUSE's edge ratio is the highest of the four variants at every one of the twelve (n, c) configurations, with no exceptions. Its SOLVED advantage is fully size-invariant at the two easier densities ($c = 4.69$: SOLVED $= 1.00$ at all four sizes; $c = 10$: likewise) and only partially so at the harder $c = 2.5$ dip, where it degrades from a small but real advantage (.08) at $n \leq 500$ to no exact solves (.00) at $n \geq 1{,}000$, though its edge ratio (.986–.991) stays comfortably above the other three throughout. Appendix J's planted Max-Cut result is a cross-distribution generalization test because its checkpoints are trained on generic, non-planted graphs. Here, training and evaluation both draw from the same planted-coloring generator with shifts in n and density, establishing within-family robustness to size and density shift.

Table 15: Linear accessibility of the graph-coloring current-color agreement margin: R^2 from a five-fold cross-validated Ridge probe applied to each configuration's hidden state, at the representative densities of Table 13.

c	OPTGNN	OPTGNN-IR	OPTGNN-LOGIT	OPTGNN-CLAUSE
1.0	.006	.302	.430	**.958**
1.5	−.015	.142	.282	**.959**
2.0	−.021	.071	.158	**.955**
2.5	−.003	.027	.086	**.931**
3.0	−.025	−.000	.050	**.938**
3.5	−.013	−.005	.046	**.960**
4.0	.012	.004	.039	**.960**
4.69	.013	.003	.037	**.952**
6.0	.121	.022	.050	**.943**
10.0	.156	.112	.140	**.943**
15.0	.328	.043	.181	**.968**
20.0	.348	.038	.256	**.987**

OPTGNN-CLAUSE's local-margin probe is both the strongest of the four and essentially flat and near-perfect ($R^2 \in [.93, .99]$) across the entire density range, including at $c = 2$–3 where OPTGNN-CLAUSE itself only solves 0–50% of instances. The margin is thus linearly decodable from its embedding regardless of whether the instance is solved. The raw edge-message aggregate already matches this probe score at representative coloring densities (Appendix E), locating the accessibility in the trained incoming messages. The result demonstrates architecture-dependent accessibility and a decodability–solving dissociation. The other three variants show a mild upward drift at high

density (OPTGNN reaching .35–.47 past $c \approx 11$) but no density range where any of them approaches OPTGNN-CLAUSE's separability.