Internal Field Report: Boolean Function Learnability (Nicolau et al., NeSy 2025)

Implications for Transformer Reasoning in SAT (Pan et al.)

Chris Esposo
Georgia Institute of Technology

September 16, 2025

Abstract

This field report summarizes my takeaways from Nicolau et al. [2025] (NeSy 2025) and triangulates those findings against our SAT–Transformer work [Pan et al., 2025]. I (i) restate results I verified in the primary text, (ii) separate what is genuinely deep from what is confirmatory for our program, and (iii) recommend concrete integration points for our Transformer–SAT pipeline (data, curricula, diagnostics, and hybridization). I include the key formulas we reference and format them to be column-safe in two-column venues.

1 Why this matters to us

Our paper proves by construction that, for bounded input sizes (p, c), a decoder-only Transformer with $O(p^2)$ parameters can implement a DPLL-like procedure via CoT and *decide* 3-SAT; we also compile this specification with PARAT and show perfect accuracy up to p=20 near the phase-transition band [Pan et al., 2025, Thm. 1.1; PARAT]. We then train Transformers end-to-end on reasoning traces: models generalize across distributions *within* the trained size range but lose accuracy beyond it—the expected "length generalization" cliff. Our datasets are deliberately constrained around $\alpha \approx 4.26$ and include *marginal* SAT/UNSAT twins to suppress shortcut signals.

Nicolau et al. [2025] ask a complementary question: how well do small MLPs learn *Boolean functions* from examples when those functions encode symbolic/combinatorial content? Their per-formula, balanced datasets (positives via near-uniform sampling; negatives via random assignments) and strict metric (100% 5-fold CV to count as "learned") produce clean signals about learnability as a function of size and constrainedness.

2 What Nicolau et al. (2025) actually show

Setup. For each formula F over n variables, build a balanced dataset of $(x, \mathbb{1}[x \models F])$ with ∼500 positives and ∼500 negatives per formula; for large or constrained formulas, positives come from UniGen2. Learn with shallow MLPs (2 layers; 200/100; ReLU vs. sigmoid) and evaluate via 5-fold CV; a formula is *perfectly learned* only if held-out accuracy is 100% [Nicolau et al., 2025].

Main results (with implications).

1. **MLP $>$ DT / Valiant** in generalization on these datasets (DTs overfit; Valiant improves with constraints but lags at low α). *Implication:* neural approximators capture Boolean concepts compactly; we should feel comfortable delegating sub-scorers to small nets while keeping symbolic checks.
2. **Small/shallow MLPs learn large encodings** (e.g., graph k-coloring, k-clique CNFs) almost perfectly. *Implication:* literal/clause heuristic heads can be very lightweight.
3. **Smaller formulas are harder** to learn perfectly (coverage sparsity and few negative patterns). *Implication:* curriculum and data coverage matter at low n.
4. **Under-constrained 3-CNFs are harder** than over-constrained; hardest-learning does *not* coincide with the SAT phase transition. *Implication:* avoid flooding training with ultra-low-α unless you handle it explicitly.

Random-3CNF macro-experiment. ∼110k formulas across $n \in \{10, 20, \ldots, 100\}$ and 11 α settings around the phase transition show: (a) strong ReLU advantage over sigmoid; (b) n=10 never reaches perfection even at 256 neurons; (c) learnability rises from very low α, peaks, then mildly drops at high α—and the critical α is *not* the minimizer. These patterns align with the four bullets above.

3 Key formulas (column-safe)

Let C_i be the i-th clause, A a (partial) assignment, and $E(\cdot)$ the standard indicator encoding. First, clause membership:

$$E(B)_v := \mathbb{1}[x_v \in B], \quad E(B)_{v+p} := \mathbb{1}[\neg x_v \in B].$$

The "not-false" and "assigned" encodings:

$$E_{\text{not-false}}(A)_v := \mathbb{1}[\neg x_v \notin A], \quad E_{\text{not-false}}(A)_{v+p} := \mathbb{1}[x_v \notin A],$$

$$E_{\text{assigned}}(A)_v = E_{\text{assigned}}(A)_{v+p} := \mathbb{1}[x_v \in A \text{ or } \neg x_v \in A].$$

Satisfaction and conflict tests (vectorized):

$$A \models F \iff \min_{i \in [c]} E(C_i) \cdot E(A) \ \geq 1, \tag{1}$$

$$F \models \neg A \iff \min_{i \in [c]} E(C_i) \cdot E_{\text{not-false}}(A) \ = 0. \tag{2}$$

Unit-propagation consequences (column-fit via `resizebox`):

$$E(D) = \max\Big\{ \min\Big(\textstyle\sum_{i=1}^{c} E(C_i)\, \mathbb{1}[E(C_i) \cdot E_{\text{not-false}}(A) = 1], 1 \Big) - E_{\text{assigned}}(A), 0 \Big\}.$$

These tests drive a parallel clause-level deduction step within a single forward pass.

4 Our existence result and compiled model

[Pan et al.] For any $p, c \in \mathbb{N}_+$ there exists a decoder-only Transformer with $O(p^2)$ parameters that autoregressively decides 3-$\mathsf{SAT}_{p,c}$ using chain-of-thought. The compiled model via PARAT achieves 100% SAT/UNSAT and valid traces up to $p{=}20$ near the 4.26 band; in practice, the longest CoT observed is $\approx 8p\,2^{0.08p}$, far below the worst-case upper bound.

5 Alignment: where Nicolau et al. helps us

Learning vs. reasoning. Nicolau learns $f(x) \in \{0, 1\}$; we learn an *algorithmic trace*. Both are sensitive to data regimes. Their per-formula balance and near-uniform positive sampling echo our own use of near-threshold distributions and marginal twins to suppress shortcuts.

Constrainedness. Their under-α difficulty cautions against naively mixing easy satisfiable cases; our training wisely fixed α near 4.26. We can now broaden to extremes *after* mastering the band.

Activation & capacity. ReLU>sigmoid for Boolean functions maps cleanly onto our use of modern non-saturating MLP activations in Transformer FFNs; their width-vs-n curves motivate scaling or curricula for length generalization.

6 What is deep vs. confirmatory

Deep: (i) *Under-constrained* is consistently harder to learn from examples; (ii) *small MLPs* nail complex encodings. **Confirmatory:** balance, per-formula evaluation, and DT/VA baselines behaving as expected.

7 Recommended integration into our pipeline

7.1 Data & curricula

1. **Constrainedness curriculum.** Train near $\alpha \in [4.1, 4.4]$ until stable; then blend bins $\{3.2, 3.5, 3.8, 4.1, 4.4, 4.7, 5.0\}$ at 10% each. *Success:* $\geq$99% SAT/UNSAT and $\geq$95% full-trace across bins; identify first failing bin.

2. **Semantic pretraining (assignment satisfaction).** Pretrain on $(F, x) \mapsto \mathbb{1}[x \models F]$ before CoT. *Success:* $>2\times$ faster CoT convergence, fewer clause-evaluation mistakes on under-constrained SAT.

3. **Length curriculum.** Expand sizes gradually (e.g., 6–10 $\rightarrow$ 6–12 $\rightarrow \cdots \rightarrow$ 6–20) while retaining earlier sizes to avoid forgetting. *Success:* smooth accuracy vs. p; no cliff at $p{+}1$.

7.2 Hybridization (light MLP modules)

1. **Clause/literal scoring head.** A 2-layer 200/100 ReLU MLP estimates $h(l \mid F)$ from static CNF features; we *gate* next-decision logits with h while the symbolic/compiled checks arbitrate. *Success:* $\geq$10% fewer CoT steps at constant exactness.

2. **Fallback at small p.** Where data coverage is sparse (small formulas), let the MLP propose top-k literals; Transformer verifies via (1)–(2).

7.3 Diagnostics & ablations

1. **Length-gen dashboard.** Track SAT/UNSAT (solid) and full-trace (dashed) vs. p; alert on drift (our Fig. 3 analogue).

2. **Activation ablation.** Swap FFN activation to sigmoid in an otherwise identical model; record convergence, trace errors.

3. **Capacity scaling.** Progressively add layers/heads when moving to larger p; warm-start from smaller model weights.

4. **Mixed-difficulty stress test.** Sprinkle 10% extreme low/high-α satisfiable formulas; monitor for shortcutting (trace validation should prevent it).

5. **Knowledge extraction.** Probes/attribution (e.g., SHAP) over the scoring head and Transformer states to surface clause/variable importance patterns.

8 Notes on unit propagation and backtracking

In our compiled/learned models, deduction uses (1)–(2) and the column-fitted unit rule. Conflicts trigger backtracking (a backjump in the abstract DPLL sense): negate the last decision literal and resume; the compiled model mirrors this behavior exactly.

9 Bottom line (my read)

Nicolau et al. mostly *confirm* our data/activation instincts while adding two practical guardrails: treat *constrainedness* as a curriculum axis, and exploit *small MLPs* for heuristic scoring under a symbolic/compiled arbiter. Together with semantic pretraining and size curricula, these changes should push our length generalization frontier and reduce reasoning steps without sacrificing guarantees.

Action items (next sprint).

1. Implement α-curriculum datasets and dashboard.

2. Add assignment-satisfaction pretraining stage before CoT.

3. Wire the lightweight scoring head (on/off ablation) with strict verification.
4. Run activation and capacity ablations; adopt the best config.

Acknowledgments

Written in my capacity as coauthor on Pan et al. [2025]; any errors are mine.

References

L. Pan, V. Ganesh, J. Abernethy, C. Esposo, and W. Lee. Can Transformers Reason Logically? A Study in SAT Solving. *arXiv:2410.07432*, 2025.

M. Nicolau, A. R. Tavares, P. H. Avelar, J. M. Flach, Z. Zhang, L. C. Lamb, and M. Y. Vardi. Understanding Boolean Function Learnability on Deep Neural Networks: PAC Learning Meets Neurosymbolic Models. In *Proc. NeSy 2025*, 2025.