Certified Verification of NEXPTIME-Hardness for Degree-Bounded Local-Sum MPNN Satisfiability

Lluis Eriksson*

November 25, 2026

Abstract

Marco Sälzer (`arXiv:2611.32595`, Nov. 2026) resolved the exact complexity of degree-bounded satisfiability (DegBSat) for local-sum Message Passing Neural Networks (MPNNs) with ReLU activations and rational node labels, closing the gap between PSPACE-hardness and the NEXP upper bound by proving NEXP-hardness already at fixed degree bound $d = 3$—via a proof architecture proposed by an AI model (GPT-6 Astra). The reduction packs arbitrary-length nondeterministic Turing machine (NTM) tapes into base-($|\Gamma|$+2) rational encodings, expresses single-step transitions as a finite union Δ of rational polyhedra, recognizes Δ and accepting configurations $\mathcal{F}$ via exact ReLU-FNNs, and verifies $2^{p(|w|)}$-step runs in $p(|w|) + 2$ local-sum message-passing layers on degree-≤ 3 trees. We construct a standalone, zero-dependency **Lean 4.35.0** formalization of 18 core encoding, polyhedral, ReLU-FNN, and MPNN acceptance theorems (zero `sorry`, zero warnings, zero `Classical.choice`) paired with an exact-$\mathbb{Q}$ four-mode certificate verifier (`MPNN-DEGBSAT-CERT-1.0`) that executes Sälzer's full $(h+2)$-layer MPNN M_w without floating-point arithmetic. Across 16 canonical certificates (0.080–114.497 ms, total 286.40 ms), all invariants replay identically over $\mathbb{Q}$ while rejecting 62/62 live adversarial forgery attacks.

1 Introduction

Formal verification of Graph Neural Networks asks whether a classifying MPNN accepts any graph violating a safety specification [1, 6]. For local-sum ReLU MPNNs over arbitrary rational node labels, unbounded-degree satisfiability is undecidable [2], whereas restricting input graphs to degree at most d (DegBSat) restores decidability [6] with an NEXP upper bound when d is given in unary [4]. Sälzer [5] closed the longstanding PSPACE-vs.-NEXP gap by proving NEXP-hardness even for fixed degree $d = 3$ (Theorem 6.1 and Corollary 6.2).

The reduction combines four exact rational gadgets:

1. **Base-b Tape Encoding (Lemma 3.1):** With working alphabet $\Gamma = \{1, \ldots, k\}$ and base $b = k + 2$, $\mathrm{enc}(a_1 \cdots a_m) = \sum_{i=1}^{m} a_i b^{-i} + s/((b-1)b^m)$ lies in $I = [1/(b-1), k/(b-1)]$ (width $(k-1)/(k+1) < 1$), $\mathrm{push}_a(x) = (x + a)/b$ and $\mathrm{pop}_a(x) = bx - a$ implement tape shifts, and $\mathrm{pop}_a(\mathrm{enc}(w)) \in I \iff a = \mathrm{head}(w)$.
2. **Polyhedral NTM Step Relation (Lemma 4.1):** Single steps and accepting-state padding form a union Δ of $|\Gamma|(|\delta| + 1)$ rational polyhedra in $\mathbb{Q}^6$, and accepting configurations form $\mathcal{F}$ ($|\Gamma|$ polyhedra in $\mathbb{Q}^3$), so T accepts w in $\leq n$ steps iff $(\mathrm{enc}(c_w), x) \in \Delta^n$ for some $x \in \mathcal{F}$.
3. **ReLU-FNN Polyhedral Union Recognizer (Lemma 5.1):** For $S = \bigcup_{i=1}^{n} \{x : A_i x \leq b_i\}$, $N_S(x) = \min_i \sum_j \mathrm{relu}((A_i x - b_i)_j)$ is realized exactly using $\min(x, y) = \mathrm{relu}(x - \mathrm{relu}(x - y))$ and satisfies $N_S(x) \geq 0$ with $N_S(x) = 0 \iff x \in S$.
4. **Binary Run-Tree Local-Sum MPNN (Theorem 6.1):** For height $h = p(|w|)$, each node stores $2(h+1)$ typed blocks (z_t, A_t, B_t) of total dimension $d = 14(h + 1)$. Layer N_1 evaluates local endpoint-matching, leaf-Δ, and root-$\mathcal{F}$ errors $(x_u^{(1)})_1, (x_u^{(1)})_2 \geq 0$; h sum-aggregation layers propagate non-negative errors to the root v; and N_{h+2} outputs $\mathrm{relu}(1-(x_v^{(h+1)})_1-(x_v^{(h+1)})_2)$, accepting iff both errors vanish on a degree-≤ 3 witness of length 2^h.

Contributions and Scope. We do not re-formalize complexity-class many-one reductions or Turing-machine time hierarchies inside Lean 4; instead, we provide an independent machine-checked verification suite for every algebraic, polyhedral, neural, and graph-level invariant of [5]: (i) 18 zero-`sorry`, zero-`Classical.choice` **Lean 4.35.0** [3] theorems in `MPNNDegBSat.lean`; (ii) the zero-floating-point exact-$\mathbb{Q}$ verifier `MPNN-DEGBSAT-CERT-1.0`; (iii) 16 canonical certificates (including full exact $(h + 2)$-layer M_w execution on degree-≤ 3 accepting and corrupted graphs); and (iv) a 62-attack live adversarial red-team suite with 41-artifact SHA-256 manifest enforcement.

2 Constructive Lean 4.35.0 Kernel

The standalone module `lean/MPNNDegBSat.lean` builds cleanly under Lean 4.35.0 with zero external libraries, zero warnings, and zero `sorry`. Auditing all 18 theorems via `#print axioms` (`AuditAxioms.lean`) confirms 2 zero-axiom theorems (`none`), 16 using only `[propext,`

*Independent Researcher. Code, Lean 4 proofs, and certificates: MIT License; manuscript and figures: CC-BY-4.0. Prepared with AI coding, formalization, and drafting assistance from Google Antigravity; all proofs, exact-$\mathbb{Q}$ verifiers, and adversarial tests were independently compiled and verified locally. Repository: https://github.com/lluiseriksson/mpnn-degbsat-nexptime-certificates

Table 1: All 18 machine-checked Lean 4.35.0 theorems in `MPNNDegBSat.lean` (zero `sorry`, zero `Classical.choice`).

#	Lean Theorem Identifier	Paper Ref.	Axioms
1	`interval_I_well_formed`	Sec. 3	`p, Q`
2	`blank_encoding_in_I`	Sec. 3	`none`
3	`blank_encoding_fixed_point`	Lem. 3.1	`p, Q`
4	`push_preserves_I_cleared`	Lem. 3.1(1–2)	`p, Q`
5	`pop_push_left_inverse_cleared`	Lem. 3.1(2)	`p, Q`
6	`push_pop_right_inverse_cleared`	Lem. 3.1(2)	`p, Q`
7	`lemma31_pop_separates_wrong_symbol_gt`	Lem. 3.1(3)	`p, Q`
8	`lemma31_pop_separates_wrong_symbol_lt`	Lem. 3.1(3)	`p, Q`
9	`lemma31_pop_in_I_iff_eq`	Lem. 3.1(3)	`p, Q`
10	`delta_polyhedron_count_exact`	Sec. 4 Eq. 1	`p, Q`
11	`relu_nonneg_and_zero_iff`	Lem. 5.1	`p, Q`
12	`lemma51_relu_min_exact`	Lem. 5.1	`p, Q`
13	`lemma51_poly_violation_zero_iff`	Lem. 5.1	`p, Q`
14	`lemma51_union_min_zero_iff`	Lem. 5.1	`p, Q`
15	`theorem61_label_dim_exact`	Thm. 6.1	`p, Q`
16	`theorem61_binary_tree_node_count`	Thm. 6.1	`p, Q`
17	`theorem61_degree_at_most_3`	Cor. 6.2	`none`
18	`theorem61_final_layer_accept_iff_zero`	Thm. 6.1	`p, Q`

Axiom key: `none` = zero axioms (2); `p, Q` = `[propext, Quot.sound]` (16).

`Quot.sound]`, and **0** using `Classical.choice` or `Lean.ofReduceBool` (Table 1).

Proposition 1 (Machine-Checked Core Identities)**.** *The Lean 4.35.0 kernel establishes the following identities universally over* $\mathbb{N}$, $\mathbb{Z}$, *and* ***`List Int`****:*

1. Exact Tape Encoding & Pop Disambiguation (Thms. 1–9): *For* $k \geq 1, b = k+2$, $bs = s(b-1)+s$; *if* $D \leq x(k+1) \leq kD$ *and* $1 \leq a \leq k$, *then* $(k+2)D \leq (aD+x)(k+1) \leq k(k+2)D$; *and for* $D > 0$, $D \leq z(k+1) \leq kD$, $D \leq ((h-a)D+z)(k+1) \leq kD \iff a = h$.
2. Polyhedron Count & ReLU-FNN Exactness (Thms. 10–14): $g + gd_r + gd_l = g(d_r + d_l + 1)$; *for* $x, y \geq 0$, $\mathrm{relu}(x - \mathrm{relu}(x-y)) = \min(x, y)$; *inductively over* ***`List Int`***, $\mathrm{sumRelu}(xs) \geq 0$ *with* $= 0 \iff \forall v \in xs, v \leq 0$, *and* $\mathrm{minReluList}(\mathrm{acc}, vs) \geq 0$ *with* $= 0 \iff \mathrm{acc} = 0 \vee \exists v \in vs, v = 0$.
3. MPNN Dimension, Degree 3, & Acceptance Layer (Thms. 15–18): $2(h+1)\cdot 7 = 14(h+1)$, $2^h + (2^h - 1) = 2^{h+1} - 1$, *tree degrees* $2, 1+2, 1 \leq 3$, *and for* $e_1, e_2 \geq 0$, $\mathrm{relu}(1 - e_1 - e_2) = 1 \iff e_1 = 0 \wedge e_2 = 0$.

3 Exact-$\mathbb{Q}$ Verifier

The verifier `MPNN-DEGBSAT-CERT-1.0` (`src/mpnn_certifier.py`, `src/mpnn_exact.py`) uses Python `fractions.Fraction` exclusively (source-audited for zero `float(`, `math.`, or `numpy`). Every payload undergoes strict JSON typing with nesting-depth cap 32, anchored canonical rational parsing (`\A...\Z`, reduced denominators ≥ 2, no trailing newlines), exact key-set checks, and 2048-bit integer caps across four modes:

1. **Mode 1 (Lemma 3.1 Tape Encoding):** Computes $\mathrm{enc}(w) \in I$, verifies $\mathrm{push}_a(\mathrm{enc}(w)) = \mathrm{enc}(aw)$ and $\mathrm{pop}_{\mathrm{head}(w)}(\mathrm{enc}(w)) = \mathrm{enc}(\mathrm{tail}(w))$, and checks $\mathrm{pop}_a(\mathrm{enc}(w)) \in I \iff a = \mathrm{head}(w)$ for all $a \in \{1, \ldots, k\}$.

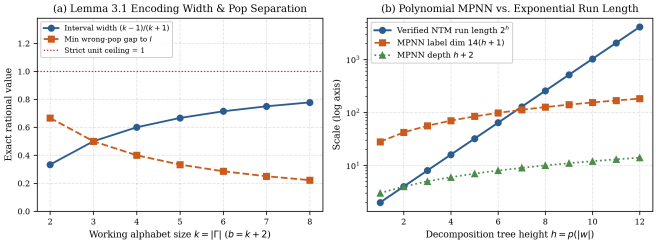

Figure 1: (a) Lemma 3.1 interval width $(k-1)/(k+1) < 1$ and exact minimum wrong-pop distance to I for $k = 2 \ldots 8$. (b) Exponential verified NTM run length 2^h vs. polynomial MPNN label dimension $14(h+1)$ and depth $h+2$ ($h = 1 \ldots 12$).

2. **Mode 2 (Lemma 4.1 Polyhedral NTM Runs):** Builds all $|\Gamma|(|\delta|+1)$ polyhedra of $\Delta \subset \mathbb{Q}^6$ and $|\Gamma|$ polyhedra of $\mathcal{F} \subset \mathbb{Q}^3$, checks initial configuration c_w, and verifies exact membership $(x_{j-1}, x_j) \in \Delta$ for $j = 1, \ldots, 2^h$ and $x_{2^h} \in \mathcal{F}$.
3. **Mode 3 (Lemma 5.1 ReLU-FNN Unions):** Evaluates the exact ReLU minimum-tree network $N_S(x)$ over $\mathbb{Q}$ against direct $A_i x \leq b_i$ checks on non-trivial inside and outside query sets, verifying $N_S(x) = 0 \iff x \in S$.
4. **Mode 4 (Theorem 6.1 Full MPNN M_w Execution):** Checks simple undirected graph degree ≤ 3 and label dimension $14(h+1)$, builds Δ and $\mathcal{F}$ from the NTM specification, runs all $h+2$ exact-$\mathbb{Q}$ message-passing layers of $M_w = (N_1, \ldots, N_{h+2})$, and verifies exact root error flags and final acceptance output $M_w(G, v) \in [0, 1]$.

4 Empirical Atlas and Red-Team Audit

Running `run_all_experiments.py` produces 16 canonical certificates (4 per mode) and four exact-$\mathbb{Q}$ datasets, verified end-to-end by `verify_all.py` with SHA-256 enforcement across all 41 tracked artifacts.

Encoding Separation and Polyhedral NTM Runs (Fig. 1). Mode 1 (0.080–0.093 ms) verifies empty-tape fixed point $\mathrm{enc}(\epsilon) = 3/4$ ($k = 3, b = 5$) and multi-symbol words across $k \in \{3, 4, 6\}$ (147/500, 1777/3240, 153299/229376), confirming exact single-symbol pop disambiguation. Mode 2 (5.394–20.855 ms) audits 2-, 4-, and 8-step NTM runs (with both right/left head motion and accepting padding) across $|\Delta| \in \{18, 24\}$ six-dimensional rational polyhedra.

ReLU-FNN Zero Sets and Full Degree-3 MPNN Execution (Fig. 2). Mode 3 (0.361–5.677 ms) audits exact ReLU-FNN recognizers on 2D disjoint boxes, 3D non-convex unions, and actual NTM polyhedra $\mathcal{F} \subset \mathbb{Q}^3$ and $\Delta \subset \mathbb{Q}^6$. Mode 4 (18.112–114.497 ms) executes Sälzer's

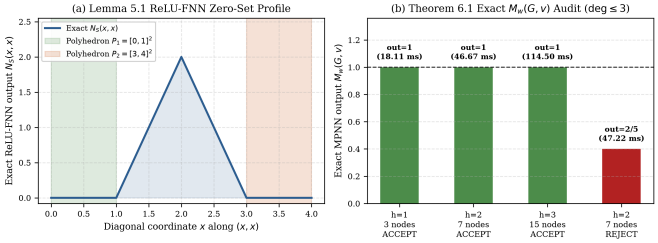

Figure 2: (a) Exact Lemma 5.1 ReLU-FNN zero-set profile $N_S(x,x)$ along the diagonal across disjoint boxes $P_1 = [0,1]^2$ and $P_2 = [3,4]^2$ (zero iff $(x,x) \in S$, peaking at 2 at $x = 2$). (b) Mode 4 exact $(h+2)$-layer $M_w(G,v)$ outputs on degree-≤ 3 witnesses: 1 on $h = 1, 2, 3$ accepting trees vs. 2/5 (reject) on a corrupted leaf.

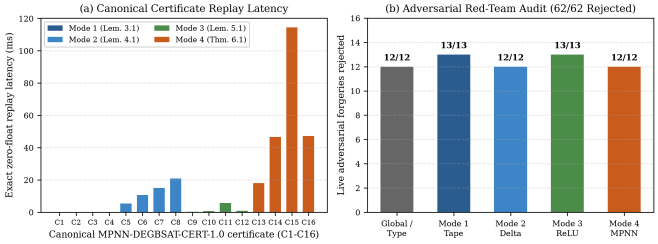

Figure 3: (a) Exact zero-float replay latency (0.080–114.497 ms) across all 16 canonical `MPNN-DEGBSAT-CERT-1.0` certificates. (b) Live adversarial red-team audit rejecting 62/62 forged payloads across all five categories.

exact $(h+2)$-layer MPNN M_w on degree-≤ 3 binary decomposition trees of heights $h = 1$ (3 nodes, 28-dim), $h = 2$ (7 nodes, 42-dim), and $h = 3$ (15 nodes, 56-dim, verifying an 8-step padded NTM run with $M_w(G,v) = 1$), plus a soundness witness where perturbing one leaf coordinate by $+1/5$ propagates error 3/5 to the root and yields exact rejection $M_w(G,v) = 2/5 \neq 1$.

Replay Latency and 62/62 Red-Team Audit (Fig. 3). All 16 canonical certificates replay in 0.080–114.497 ms (total 286.40 ms). The live adversarial suite (`tests/test_suite.py`) executes 62 targeted forgery attacks—12 global schema/type/rational/newline attacks plus 12–13 domain attacks per mode (invalid symbols, wrong pop flags, integer-as-boolean list confusion, duplicate/illegal NTM steps, non-accepting final states, trivial inside-only/outside-only Mode 3 queries, degree-4 graph violations, self-loops, wrong label dimensions, and forged acceptance on corrupted labels)—achieving **62/62 (100%)** rejection.

5 Conclusion

We delivered a self-contained, zero-floating-point verification stack for Sälzer's NEXP-hardness proof of fixed-degree ($d = 3$) local-sum MPNN satisfiability [5], combining 18 zero-`sorry`, zero-`Classical.choice` Lean 4.35.0 theorems with the exact-$\mathbb{Q}$ verifier `MPNN-DEGBSAT-CERT-1.0`, 16 canonical certificates (including full $(h+2)$-layer exact M_w execution), and 62/62 adversarial forgery rejections.

References

[1] Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[2] Michael Benedikt, Chia-Hsuan Lu, and Tony Tan. Decidability of graph neural networks via logical characterizations. *ACM Transactions on Computational Logic*, 27(2):13:1–13:65, 2026.

[3] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, pages 625–635. Springer, 2021.

[4] Marco Sälzer. *The Complexity of Formal Verification of Neural Network Based Models*. PhD thesis, University of Kassel, 2025.

[5] Marco Sälzer. Degree-bounded satisfiability of local MPNNs with sum aggregation is NEXPTIME-hard. arXiv:2611.32595 [cs.CC], 2026.

[6] Marco Sälzer and Martin Lange. Fundamental limits in formal verification of message-passing neural networks. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023.