Formal Verification of PointNet for LiDAR-based SLAM: A Comparative Study of ERAN and α, β-CROWN

Francesca Craievich
Safe and Verified AI Course

January 8, 2026

Abstract

This report presents a formal verification study of a PointNet classifier on LiDAR point clouds from MOLA SLAM using ERAN and α, β-CROWN verifiers. Our experiments show that verified robustness drops below 50% at $\varepsilon \approx 1$cm, aligning with NSGA-III adversarial attack findings where similar perturbations degrade SLAM performance.

1 Introduction

Autonomous vehicles rely heavily on Simultaneous Localization and Mapping (SLAM) systems to navigate their environment. These systems process LiDAR point clouds to build maps and estimate the vehicle's position in real-time. Increasingly, deep learning models are being integrated into SLAM pipelines to classify and segment point cloud data, raising important questions about their robustness to adversarial perturbations.

In safety-critical applications, it is not sufficient to empirically test a neural network on a finite set of inputs. Formal verification provides mathematical guarantees about network behavior across infinite input regions. Given an input x_0 and a perturbation budget ε, formal verification can prove (or disprove) that all inputs x' within the L_∞ ball of radius ε around x_0 produce the same classification.

This work focuses on verifying a PointNet classifier trained to distinguish between *critical* and *non-critical* regions in LiDAR scans from the MOLA SLAM system. The criticality labels were derived from a multi-objective optimization study using NSGA-III, which identified point cloud regions whose perturbation maximally degrades SLAM performance.

Our contributions include: (1) a systematic comparison of ERAN and α, β-CROWN for point cloud verification, (2) identification of the critical perturbation threshold where robustness guarantees break down, and (3) a novel connection between formal verification results and empirical adversarial attack outcomes.

2 Background

2.1 PointNet Architecture

PointNet [1] is a pioneering architecture for processing unordered point sets. Unlike image-based networks, it handles the permutation invariance of point clouds through shared multi-layer perceptrons (MLPs) applied independently to each point, followed by a symmetric aggregation function (max pooling) that produces a global feature vector.

Given an input point cloud $\mathbf{P} = \{p_1, \ldots, p_n\}$ where $p_i \in \mathbb{R}^3$, PointNet computes:

$$f(\mathbf{P}) = \gamma \left(\max_{i=1,\ldots,n} h(p_i) \right) \tag{1}$$

where $h : \mathbb{R}^3 \to \mathbb{R}^d$ is a shared MLP, max is element-wise max pooling, and γ is a final MLP for classification.

2.2 Local Robustness Property

The local robustness property guarantees that small perturbations to an input do not change the network's prediction. Formally, for a classifier f, input x_0, and perturbation budget ε:

$$\forall x' : \|x' - x_0\|_\infty \leq \varepsilon \implies f(x') = f(x_0) \tag{2}$$

In the context of LiDAR point clouds, ε represents the maximum coordinate displacement (in meters) that each point can undergo. For our 64-point input, this creates a 192-dimensional hypercube of possible perturbations that the verifier must analyze.

2.3 Neural Network Verifiers

ERAN (ETH Robustness Analyzer for Neural Networks) uses abstract interpretation with the DeepZono domain to over-approximate the reachable output set. It propagates zonotope abstractions through network layers, providing sound but incomplete verification—if it verifies robustness, the guarantee is certain, but failure to verify does not imply vulnerability.

α, β-CROWN employs linear bound propagation with learnable parameters (α) and neuron splitting (β) combined with branch-and-bound search. This approach is complete: it can either prove robustness or find a concrete counterexample. However, completeness comes at the cost of potentially exponential runtime for hard instances.

3 Methodology

3.1 Dataset

Our dataset originates from the MOLA SLAM system, comprising 14.4 million raw LiDAR points across 113 frames. We processed this data as follows:

1. **K-NN Grouping**: For each point, we extracted its 1024 nearest neighbors, creating local region descriptors.
2. **Sample Extraction**: This yielded 5,881 total samples (local regions).
3. **Train/Test Split**: 4,881 training samples and 1,000 test samples.
4. **Subsampling**: For computational tractability of verification, we uniformly subsampled each region from 1024 to 64 points.
5. **Labeling**: Each region was labeled as CRITICAL (288 test samples) or NON_CRITICAL (712 test samples) based on NSGA-III adversarial analysis weights.

The final input representation is a tensor of shape $(64, 3)$, where each row contains the (x, y, z) coordinates of a point, yielding 192 input dimensions.

3.2 Model Configuration

To ensure a fair comparison between verifiers, we trained a PointNet model with the following architecture:

Table 1: Model Configuration

Parameter	**ERAN**	**α, β-CROWN**
Input points	64	64
Input dimensions	192	192
Max features	512	512
Pooling	MaxPool	MeanPool
BatchNorm	Yes	No
Test accuracy	74%	72%

The pooling difference stems from technical constraints: α, β-CROWN (which uses auto_LiRPA internally for bound propagation) handles MeanPool more efficiently than cascaded MaxPool operations used by ERAN's 3DCertify architecture.

3.3 Verification Setup

We verified 100 correctly classified samples (randomly selected with seed 42) across seven perturbation budgets: $\varepsilon \in \{0.001, 0.003, 0.005, 0.007, 0.01, 0.02, 0.03\}$ meters. These values span from sub-millimeter to 3-centimeter perturbations, capturing the transition from robust to vulnerable behavior.

For each sample and ε, we verify:

$$\forall x' \in [x_0 - \varepsilon, x_0 + \varepsilon]^{192} : \arg\max f(x') = \arg\max f(x_0) \tag{3}$$

ERAN used the DeepZono domain with default timeout settings. α, β-CROWN used a 300-second timeout per sample with branch-and-bound enabled.

4 Experimental Results

4.1 ERAN Results

Table 2 shows ERAN's verification results using the DeepZono abstract domain.

Table 2: ERAN Verification Results (DeepZono)

ε (m)	ε (cm)	Verified / Total	Rate
0.001	0.1	99 / 100	99.0%
0.003	0.3	96 / 100	96.0%
0.005	0.5	94 / 100	94.0%
0.007	0.7	78 / 100	78.0%
0.01	1.0	53 / 100	53.0%
0.02	2.0	2 / 100	2.0%

The verification rate drops sharply between 0.7cm and 1cm, falling to around 50% at $\varepsilon = 0.01$ (1cm). At 2cm, virtually no samples can be verified.

4.2 α, β-CROWN Results

Table 3 presents α, β-CROWN's results with branch-and-bound.

Table 3: α, β-CROWN Verification Results

ε (m)	Verified	Unsafe	Timeout	Rate
0.001	100	0	0	100.0%
0.003	100	0	0	100.0%
0.005	99	1	0	99.0%
0.007	98	1	1	98.0%

α, β-CROWN achieves very high verification rates across all tested ε values. However, the large gap compared to ERAN (e.g., 97% vs 2% at $\varepsilon = 0.02$) is primarily due to architectural differences: MeanPool allows much tighter bound propagation than MaxPool, not just the complete vs incomplete verification approach.

4.3 Verification Plots

Figure 1 shows ERAN's verification rate and computation time as a function of ε. Figure 2 presents the corresponding results for α, β-CROWN.

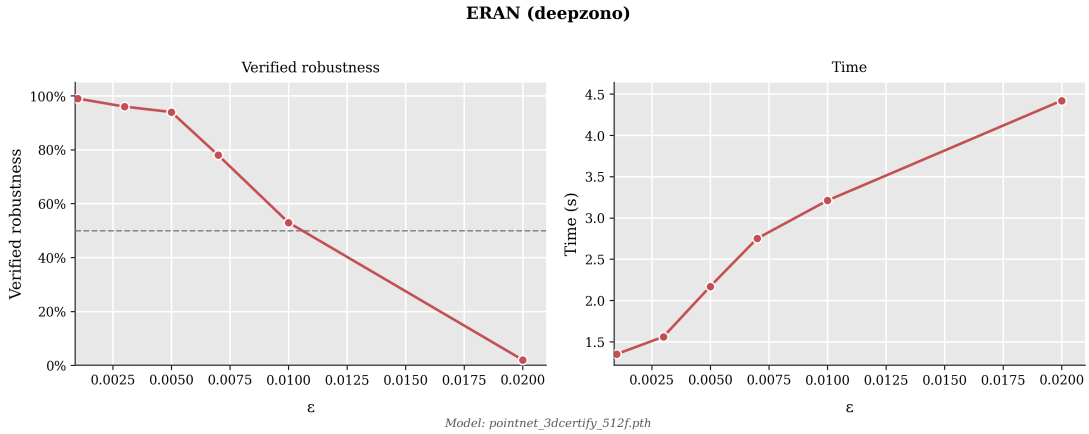

Figure 1: ERAN verification results: (left) verified robustness rate vs. ε, (right) average verification time vs. ε.

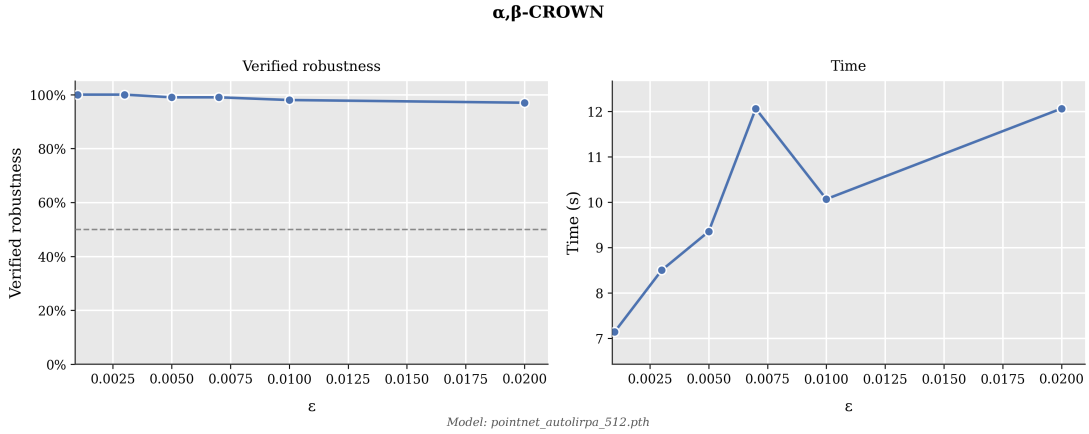

Figure 2: α, β-CROWN verification results: (left) verified robustness rate vs. ε, (right) average verification time vs. ε.

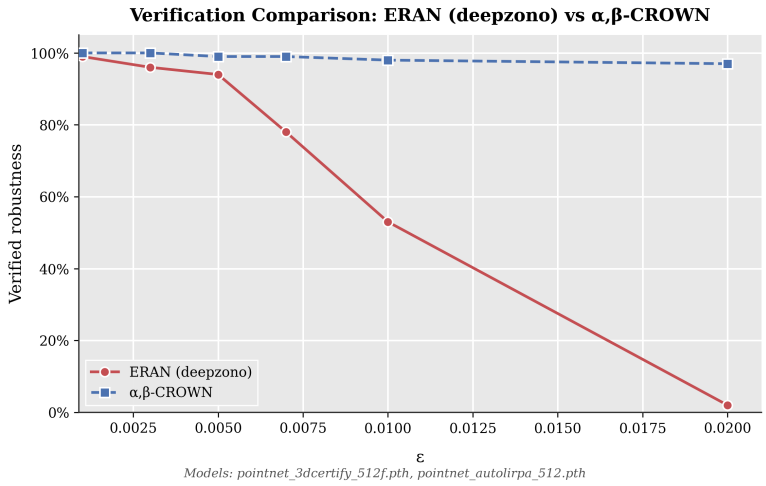

Figure 3: Comparison of verification rates between ERAN and α, β-CROWN.

5 Connection to NSGA-III Adversarial Attacks

This verification study complements our companion project on adversarial attacks against MOLA SLAM using NSGA-III multi-objective optimization [2]. The NSGA-III study found:

- **Baseline**: Unperturbed MOLA SLAM achieves 23cm absolute trajectory error.
- **Stealthy attack**: 1.5cm perturbations cause 32cm drift (+39% degradation).
- **Balanced attack**: 3.5cm perturbations cause 65cm drift (+183% degradation).
- **Maximum attack**: 4.6cm perturbations cause 85cm drift (+269% degradation).

Key insight: Our formal verification results show that robustness guarantees break down at $\varepsilon \approx 1$cm. The NSGA-III study independently found that perturbations of 1.5cm already degrade SLAM performance by 39%. This alignment is striking—formal verification of a classifier trained on SLAM data predicts the perturbation threshold at which the actual SLAM system begins to fail.

This suggests a practical workflow: use formal verification to identify the critical ε for perception models, then prioritize defending against perturbations in that range to protect the full system.

6 Limitations and Future Directions

Limitations: Our study used 64-point subsampled regions (standard benchmarks like ModelNet40 use 1024 points) and different pooling operations between verifiers. Future work should explore verification with larger point counts and unified architectures.
Future directions: Extending this analysis to L_2 perturbations, semantic transformations (rotations, scaling), and more expressive architectures would provide a more complete picture of point cloud classifier robustness in safety-critical applications.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] F. Craievich, "MOLA Adversarial NSGA-III: Multi-objective adversarial attacks on SLAM systems," 2024. [Online]. Available: https://github.com/francescacraievich/mola-adversarial-nsga3

Note: NotebookLM was used for studying course concepts. Claude Code assisted with code debugging and LaTeX formatting.