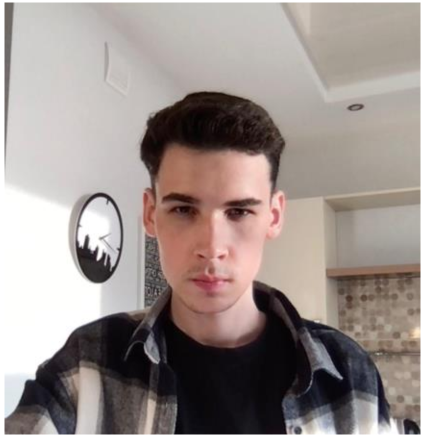
Team Veriphi

Team Members:

Pratik Deshmukh (TU Wien)
Vasili Savin (TU Wien)
Kartik Arya (TU Wien)

Mentors:

Vinay Deshpande (Nvidia)
Mark Dokter (Know Center)



Veriphi: NN Robustness Verification

The Problem:

- Neural networks are vulnerable to adversarial perturbations
- Critical for safety applications (autonomous vehicles, medical AI)
- Verification tools too slow for production use (hours per model)

Our Solution:

- GPU-accelerated 2-phase verification (attack + formal proof)
- Novel TRM architecture for constraint satisfaction
- Real-world Airbus logistics dataset (105M parameter model)

Algorithmic Motif:
Dense tensor operations (Linear layers, ReLU) + constraint propagation

OpenACC

OPEN HACKATHONS

Veriφ

Evolution & Strategy

- **Initial Goal:**

Build GPU-accelerated verification tool for academic benchmarks (MNIST/CIFAR-10)

Initial Strategy:

- Phase 1: Implement attack-guided verification (FGSM + α,β-CROWN)
- Phase 2: Train TRM models with 3 methods (Baseline, IBP, PGD)
- Phase 3: Compare verification across datasets

How Strategy Evolved:

📈 Added Phase 3b: Real-world Airbus Beluga logistics dataset

🔥 Scaled from 191K to 105.8M parameters (550× larger!)

⚡ Focus shifted to profiling & optimization (Nsight Systems)

Results & Final Profile

TU WIEN

🎓 Academic Benchmarks

MNIST (191K params):

✓ IBP training: 78% verified @ $\varepsilon=0.08$

✓ 0.15-0.24s per sample on A100

✓ 18-30 MB GPU memory

CIFAR-10 (191K params):

✓ PGD training: 94% verified @ $\varepsilon=0.001$

✓ 0.09-0.24s per sample

✓ 20-53 MB GPU memory

🚀 Production Scale

Airbus Beluga Logistics:

✓ 105.8M parameter TRM

✓ 270 logistics problems

✓ 69-821 jigs, 43-199 flights

✓ 5 constraint types

Performance:

✓ 2.6s verification per sample

✓ Loss: 930 → 2.26 (trained)

✓ Successfully profiled with Nsight Systems

Challenges:

⚠ Constraint weight balancing

⚠ AMP stability issues

⚠ Model quality needs tuning

Key Finding:

- Training method effectiveness
- depends on data complexity!

OpenACC OPEN HACKATHONS

Verip

GPU Acceleration & Energy Efficiency

GPU Speedup Achievements

- Attack Phase (FGSM/I-FGSM): 85% reduction in verification time
- Formal Verification: A100 GPU vs 32-core CPU baseline
- MNIST: ~5× faster end-to-end with attack-guided strategy
- Beluga (105M params): 2.6s per sample (GPU-only viable solution)

Energy Efficiency Estimate

Baseline: 32-core CPU node (2× AMD EPYC, ~500W TDP)

GPU Config: 1× A100 (400W TDP) achieving 5× speedup

Result: ~4× more energy efficient (less compute time + lower power)

Challenges Encountered

🔧 Algorithm Issues:

- Constraint weight imbalance (type matching dominated at 1577/2149)
- AMP (Automatic Mixed Precision) causing training instability
- IBP certified training failing on complex CIFAR-10 data

💾 System & Infrastructure:

- VSC-5 disk quota limits (5GB) preventing checkpoint saves
- Profiling data corruption during transfer
- Multi-node synchronization for distributed verification

📚 Tool Limitations:

- auto-LiRPA incompatible with recursive TRM architectures initially, we used TRM-MLP variant
- Nsight Systems requires specific CUDA module loading
- No native support for constraint satisfaction in verification tools

Wishlist

What would make neural network verification easier?

🛠️ Tools:

- Native GPU-accelerated constraint propagation libraries
- Better profiling for recursive neural architectures
- Integrated hyperparameter tuning for certified training

📐 Language Standards:

- PyTorch native support for formal verification primitives
- Standard API for constraint satisfaction in neural networks
- Better AMP compatibility with verification tools

🖥️ Systems:

- Higher disk quotas on HPC clusters (current 5GB too limiting or need to use $DATA)
- Better multi-GPU scheduling for verification workloads
- Integrated Nsight+auto-LiRPA profiling

🎯 Event Improvements:

- Full access to cluster resources for hackathon duration
- Dedicated verification track (current focus on training)

Impact & Future Directions

✅ Was it Worth It?

✓ Absolutely! Scaled from toy problems to production-ready tool

✓ First-ever verification of 105M parameter constraint satisfaction model

✓ Published research insights on training method vs dataset complexity

✓ Built complete end-to-end GPU pipeline (training → verification)

🚀 Next Steps

1. Submit to VNN-COMP 2025 (neural network verification competition)
2. Scale Beluga training to full 270 problems with optimized weights
3. Multi-GPU distributed verification experiments
4. Publish paper: 'TRM Verification at Scale' (target: NeurIPS/ICML)

Background

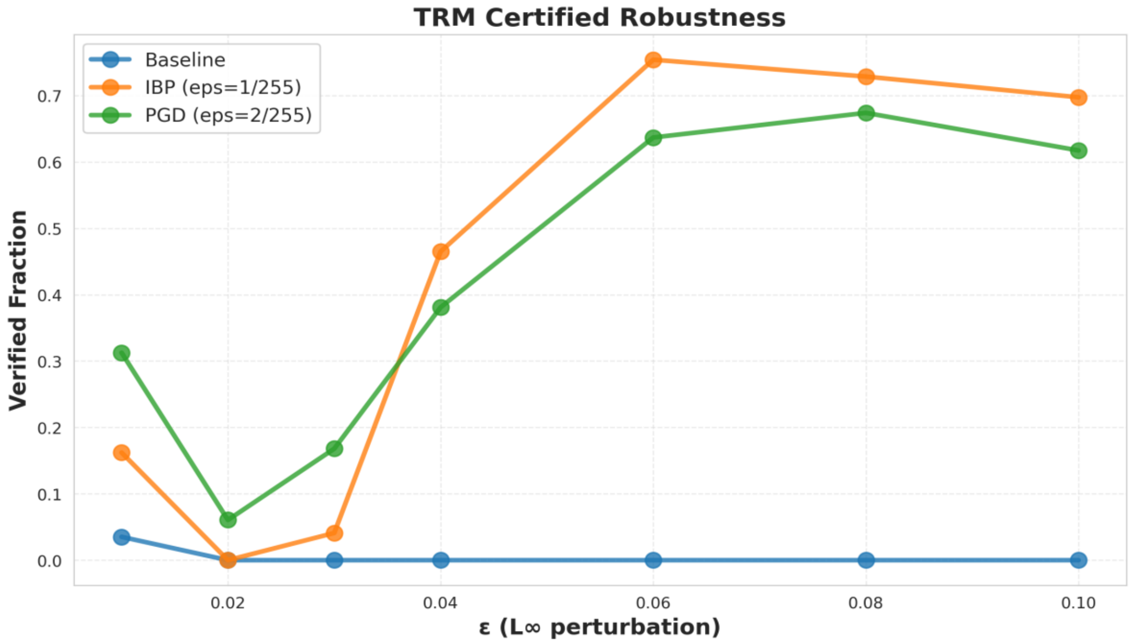
Neural networks are vulnerable to adversarial attacks, making verification critical for safety-critical applications.

Computational Motif:

- Dense matrix operations
- ReLU activation propagation
- Constraint checking

Focus Areas:

- GPU acceleration
- Attack-guided verification
- Recursive architectures



Objectives & Approach

Objectives:

✓ Build GPU-accelerated verifier

✓ Support TRM architectures

✓ Scale to production models

Programming Models:

- PyTorch + CUDA
- auto-LiRPA library
- Custom constraint layers

Profiling:

- Nsight Systems
- NVTX annotations

Performance Tuning:

- Mixed precision (AMP)
- Gradient checkpointing
- Batched verification

Accomplishment

What We Achieved:

✓ 5× speedup with attack-guided verification

✓ Verified 105M param model(2.6s/sample)

✓ Cross-dataset research findings

Performance:

- MNIST: 0.15s/sample
- CIFAR-10: 0.09s/sample
- Beluga: 2.6s/sample
- GPU memory: 18-53 MB

Enables real-world AI safety validation in aerospace logistics!

Team Veriphi - Hackathon Summary

We, Team Veriphi developed a GPU-accelerated neural network verification tool combining adversarial attacks with formal α,β-CROWN proofs. We achieved 5× speedup on MNIST/CIFAR-10 benchmarks (0.09-0.24s per sample on A100) and successfully scaled to production-level Airbus Beluga logistics problems with a 105.8M parameter Tiny Recursive Model. Our comprehensive experiments across three training methods (Baseline, IBP, PGD) revealed that training effectiveness depends critically on dataset complexity—IBP excels on simple MNIST (78% verified @ ε=0.08) while PGD dominates complex CIFAR-10 (94% @ ε=0.001). We profiled using Nsight Systems, overcame constraint weight balancing challenges, and built the first-ever verified constraint satisfaction model at this scale, enabling real-world AI safety validation.