Bit-Precise Neural Network Verification

Edoardo Manino

Lecturer in AI Security

14 January 2026

Table of Contents

Adversarial attacks

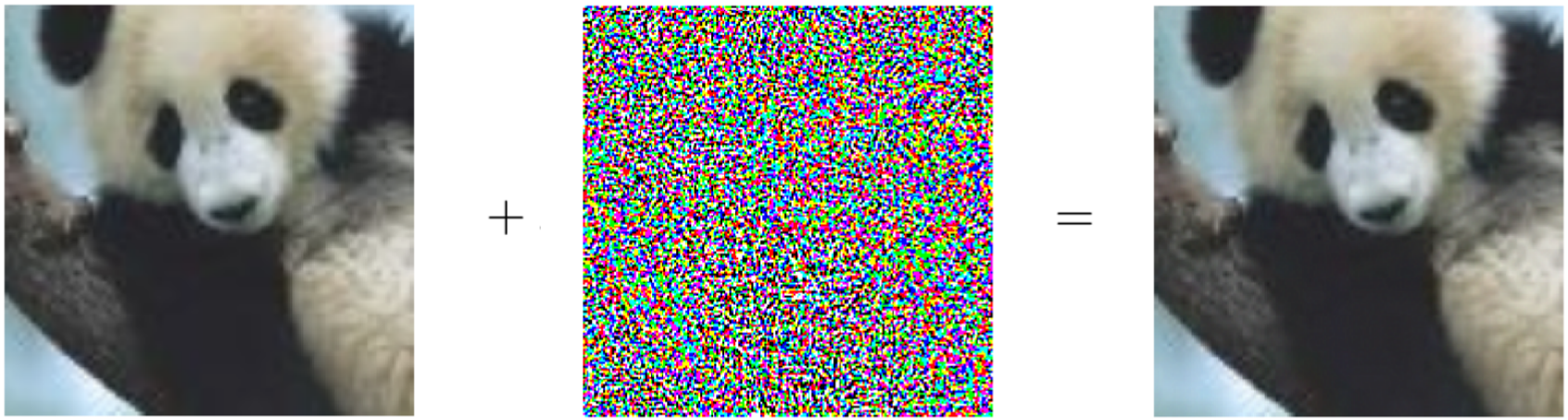
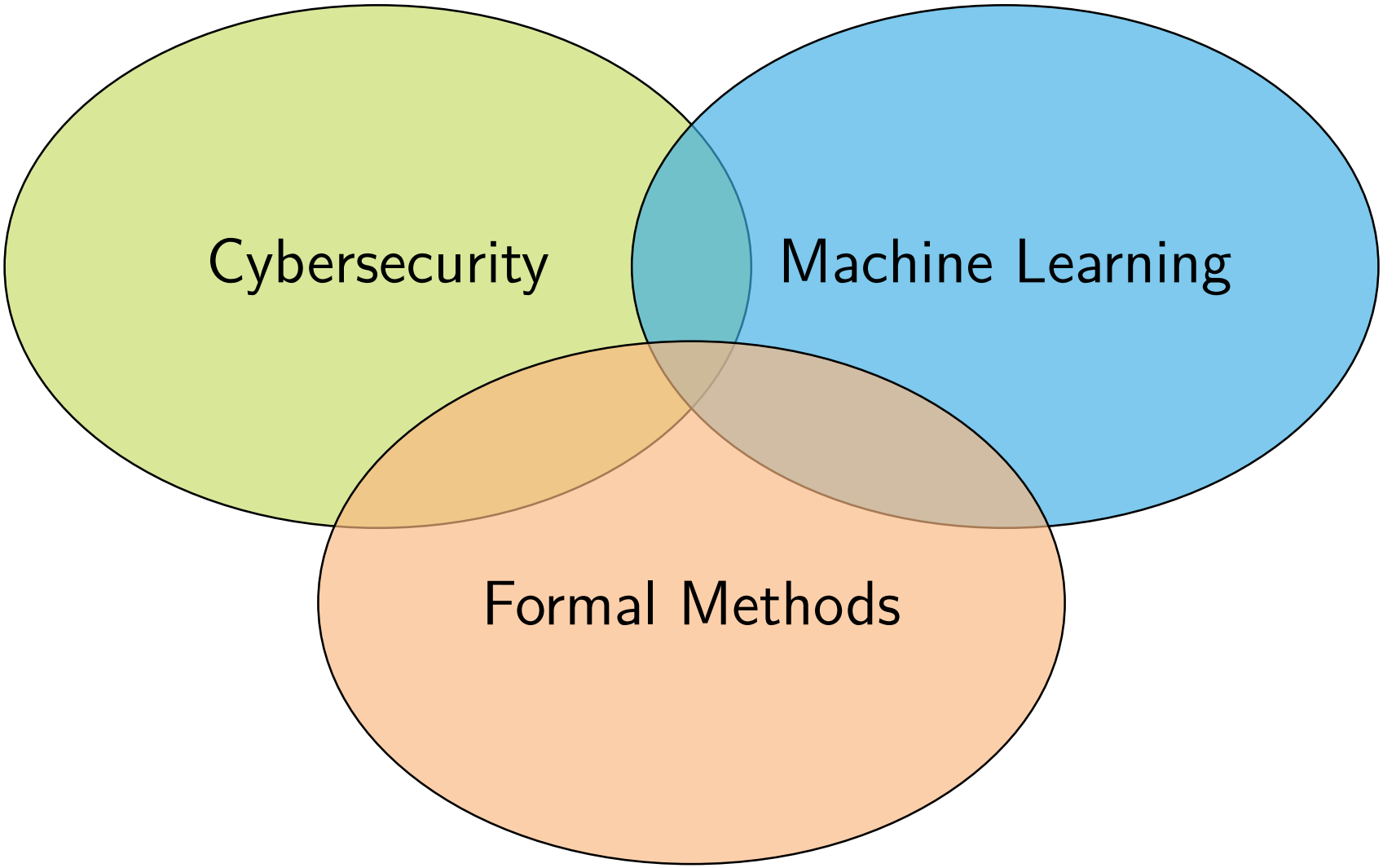


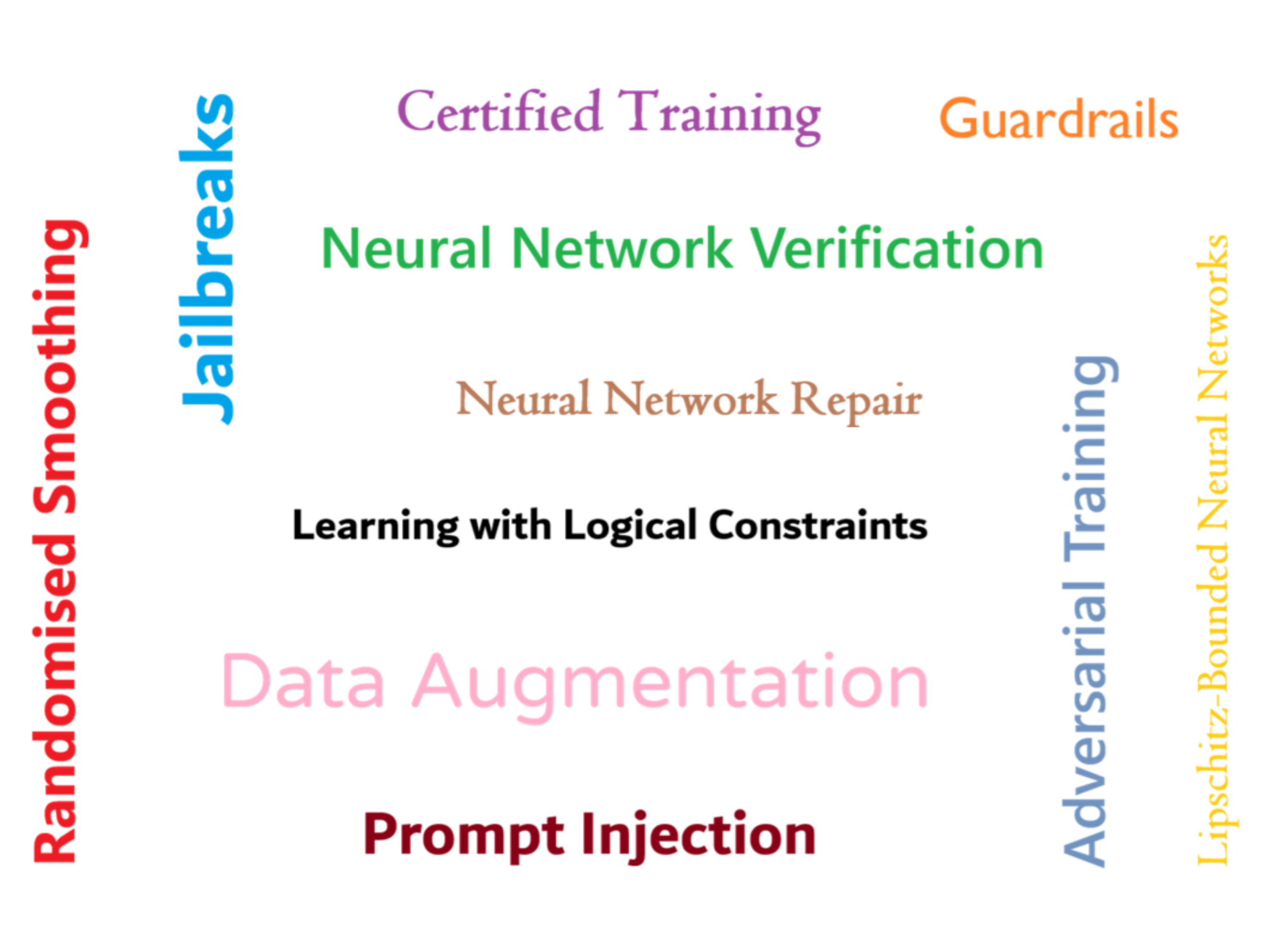
Figure: Image adapted from Goodfellow *et al.*, 2015.

Neural networks are *not* robust to input noise

- $\exists x', \quad ||x - x'||_\infty \leq \epsilon \quad \not\Longrightarrow \quad f(x) = f(x')$

Fertile Ground for Research




Randomised Smoothing
Jailbreaks
Certified Training
Guardrails
Neural Network Verification
Neural Network Repair
Learning with Logical Constraints
Data Augmentation
Prompt Injection
Adversarial Training
Lipschitz-Bounded Neural Networks

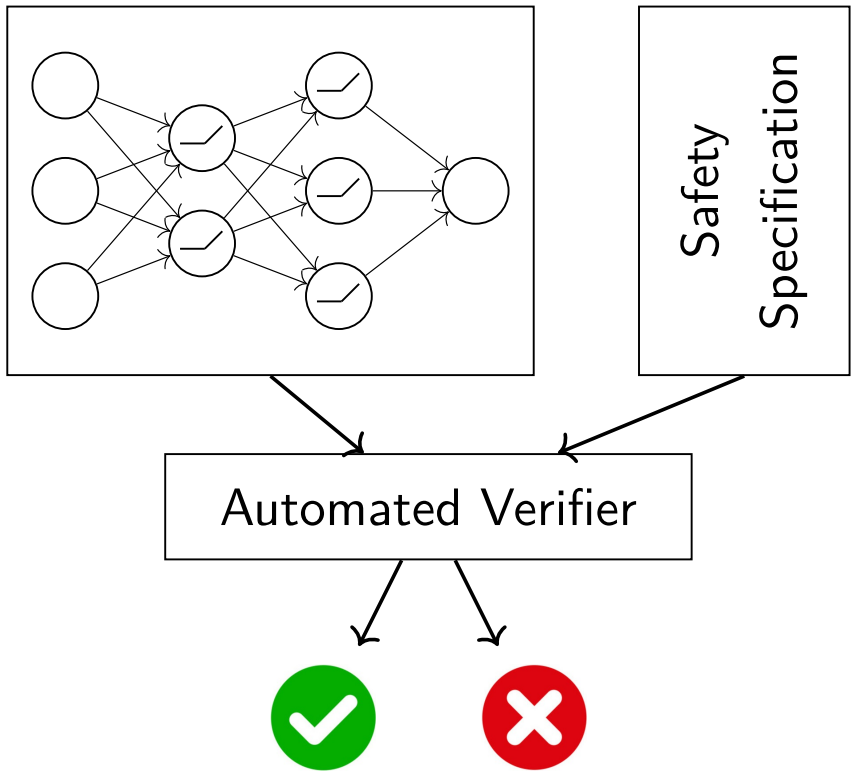
Neural Network Verification (1)

Common use cases:

- Hardware
- Software
- CPS

And now...

- Neural networks!



We "just" need to agree on

- Which properties and specifications we care about
- What are scalable algorithms and sensible use cases

Neural Network Verification (2)

Specifications as pre- and post-conditions

- $\forall x \in \mathcal{I}n, \quad f(x) \in \mathcal{O}ut$
- Includes robustness, monotonicity, equivalence, stability, ...

Algorithms combine

- Reachability, aka set propagation
- Optimisation, aka encoding to a solver
- Search, aka splitting into sub-problems

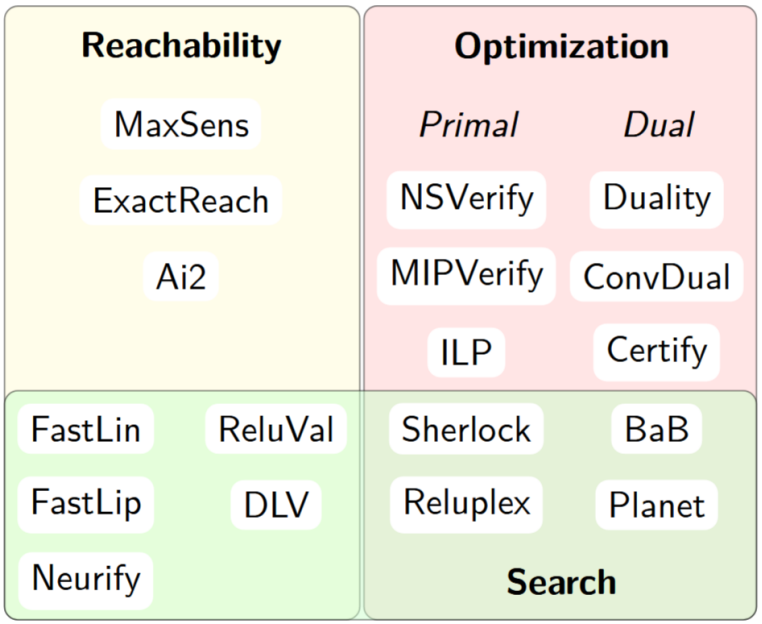
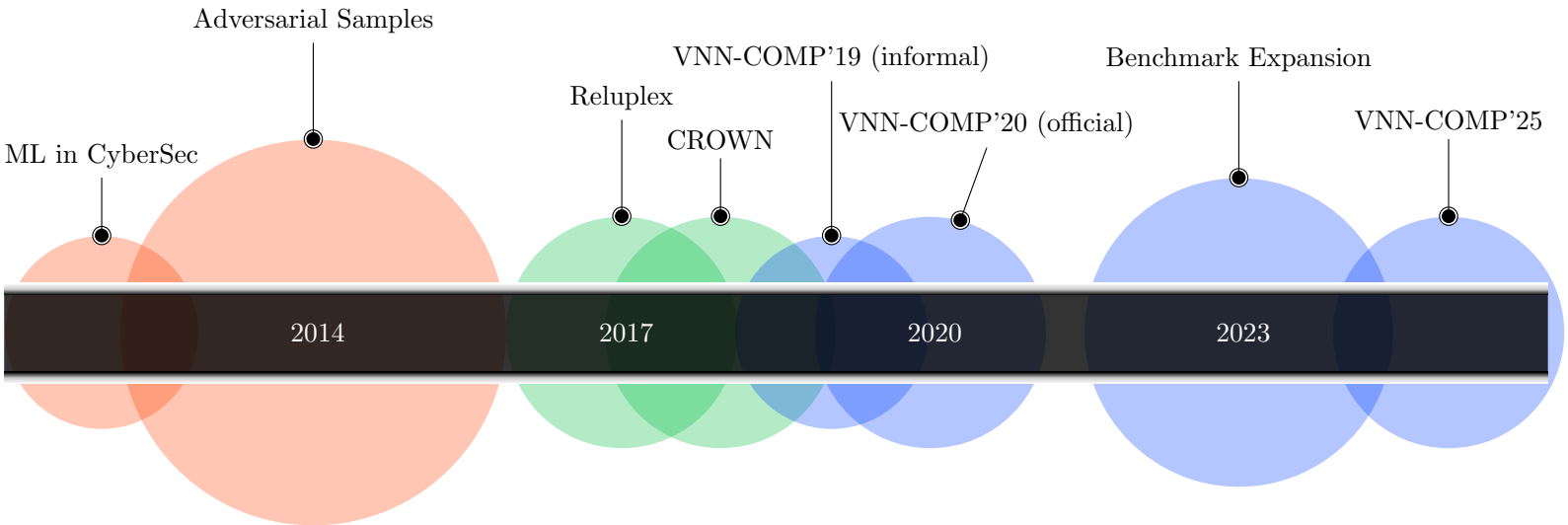


Figure: Image by Liu *et al.*, 2021

Neural Network Verification: a Timeline



See report:

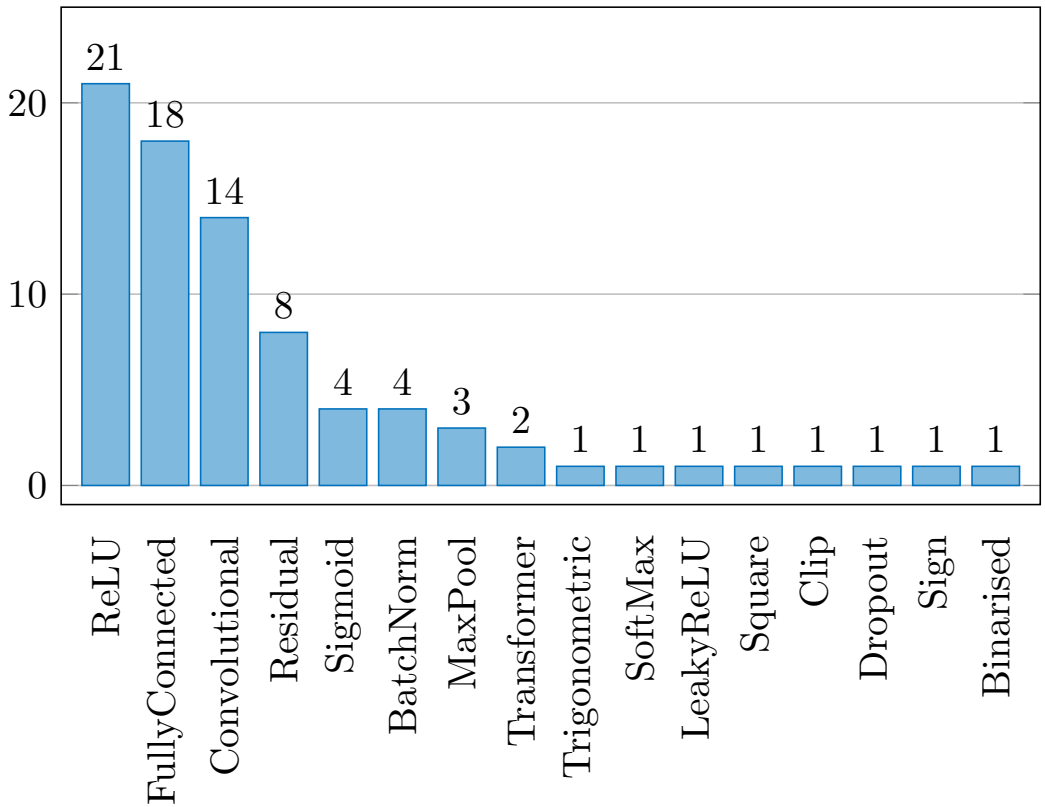
- The 6th International Verification of Neural Networks Competition (VNN-COMP 2025): Summary and Results

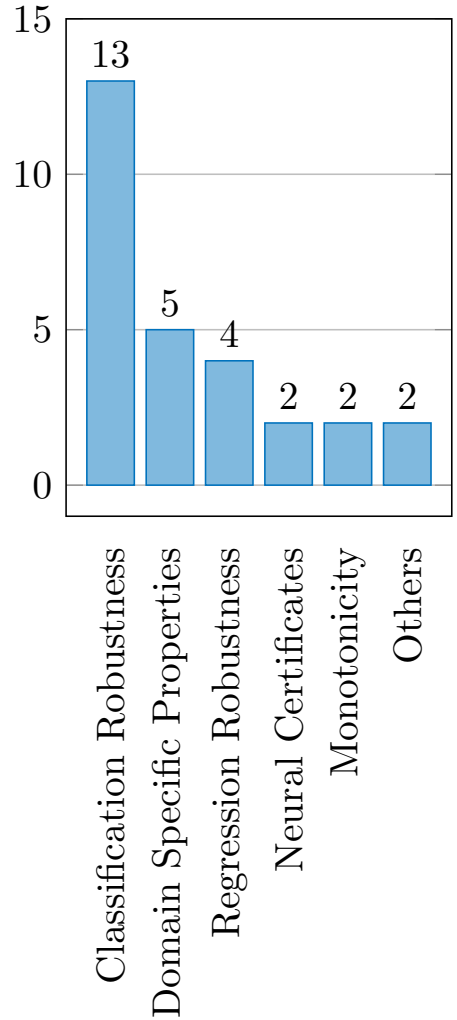
VNN-COMP Benchmark Size

Benchmark	*# Params*
VGGNet16	138M
cGAN	500k - 68M
TLL Verify Bench	17k - 67M
NN4Sys	33k - 37M
Yolo	22k - 37M
Metaroom	466k - 7.4M
cifar100	2.5M - 3.8M
tinyimagenet	3.6M
Collins Aerospace	1.8M
SoundnessBench	1.7M
Traffic Signs	905k - 1.7M
MalBeWare	102k - 1.5M
CORA	575k - 1.1M

Benchmark	*# Params*
Dist Shift	342k - 855k
ml4acopf	4k - 680k
Relusplitter	13k - 625k
Collins RUL	60k - 262k
LinearizeNN	203k
CCTSDB	100k
ViT	68k - 76k
SAT ReLU	100 - 35k
Acas XU	13k
safeNLP	4k
cersyve	1k - 4k
LSNC-ReLU	275

VNN-COMP Benchmark Features





Find all benchmarks at: `https://sites.google.com/view/vnn2025`

VNN-COMP @ AAAI 2026 Lab Forum

LH03: The Verification of Neural Networks Competition (VNN-COMP): A Lab for Benchmark Proposers, Verification Tool Participants, and the Broader AI Community

Lab Code: LH03 **Date:** Wednesday, January 21, 2026 **Time:** 8:30am - 12:30pm (SGT) **Location:** Singapore

AAAI Tutorial & Lab Forum | Official Lab Listing

Organizers: Taylor T. Johnson, Edoardo Manino, ThanhVu Nguyen, Christopher Brix, Konstantin Kaulen, Changliu Liu, Ziwei Wang, Matthew L. Daggitt

With additional contributions from Stanley Bak, Tobias Ladner, and other VNN-COMP organizers

See the full schedule at: `https://vnn-comp.github.io/#aaai2026`

Example 1: Neural Network Compression

Collaboration with:

- Universidade Federal do Amazonas

Partially funded by:

- Engineering and Physical Sciences Research Council (EPSRC)

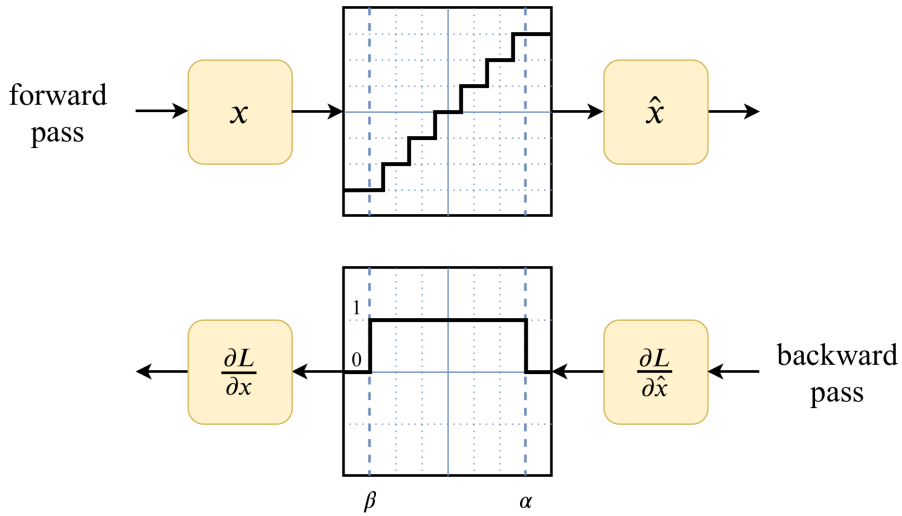
See papers:

- Counterexample-Guided Neural Network Quantization Refinement
- Towards Global Neural Network Abstractions with Locally-Exact Reconstruction

Neural Network Quantization

Many Strategies

- Dynamic
- Post-Training
- Q-Aware Training
- Non-Uniform
- ...



Main differences

- Whether the weights and/or the activations are quantized
- Whether the weights are fine-tuned after quantization
- Whether the quantization is uniform (e.g., int 8-bit)

Quantisation and NN Equivalence

		Number of bits												
Safety Prop.		6	7	8	9	10	11	12	13		28	29	30	31	32
Set.	R_{40}	S	S	F	S	S	S	S	S	...	S	S	S	S	S
	R_{50}	S	S	F	F	F	F	F	F	...	F	F	F	F	S
Vers.	R_{20}	S	F	S	S	S	S	S	S	...	S	S	S	S	S
	R_{30}	S	F	S	S	S	S	S	S	...	S	S	S	S	S
	R_{40}	S	F	S	F	F	F	S	S	...	S	S	S	S	S
	R_{50}	S	F	F	F	F	F	F	F	...	F	F	F	F	F
Virg.	R_{20}	S	F	S	S	S	S	S	S	...	S	S	S	S	S
	R_{30}	S	F	S	S	S	S	S	S	...	S	S	S	S	S
	R_{40}	S	F	S	S	F	S	S	S	...	S	S	S	S	S
	R_{50}	S	F	F	F	F	F	F	F	...	F	F	F	F	F

Table: Effects of quantization on the robustness of a NN trained on Iris data.

Effects of Quantisation

- Even if the accuracy does not drop, the behaviour may change
- Can we deploy *safe* quantized network?

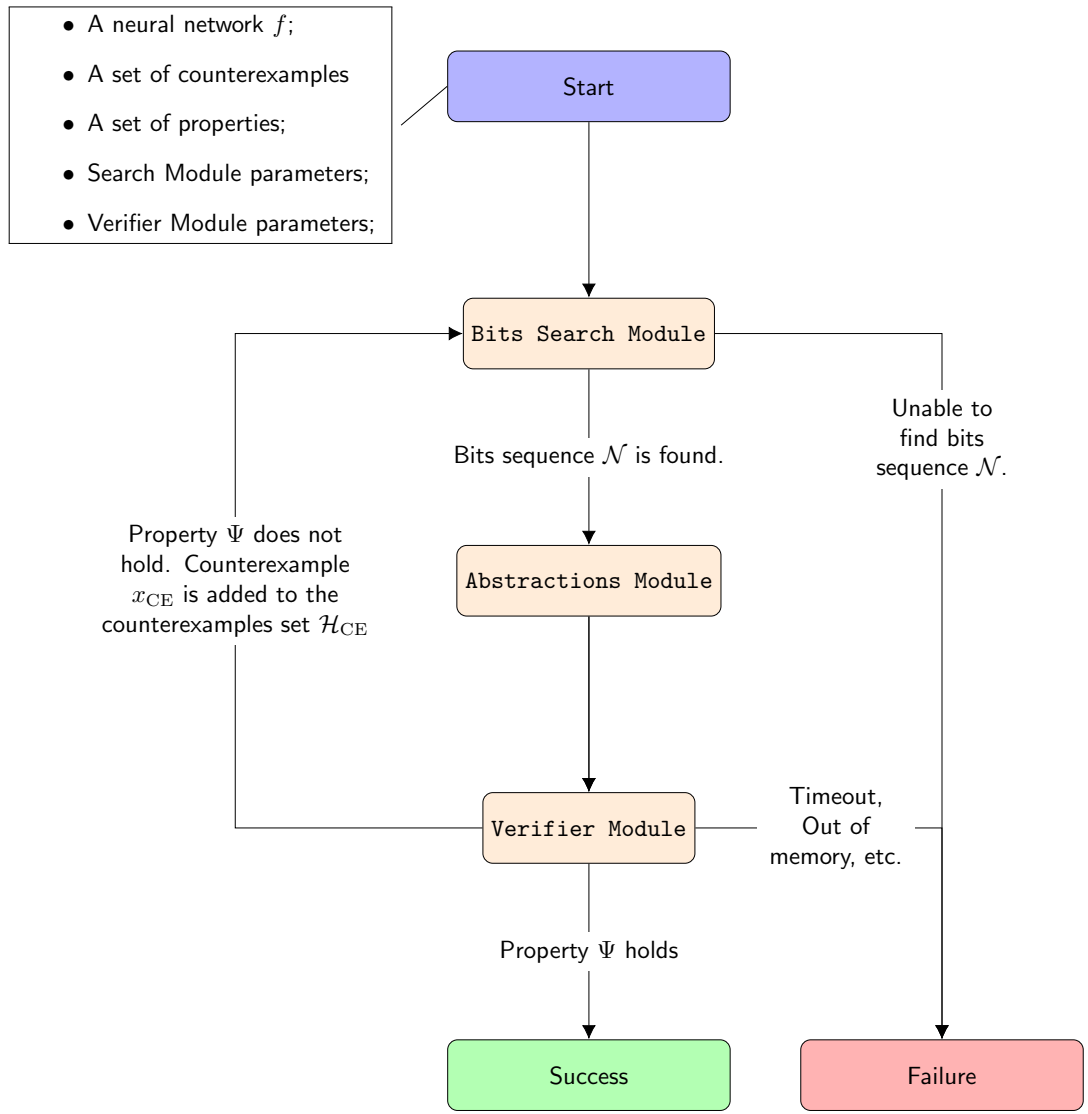
CEG4N: Counterexample-Guided NN Quantisation (1)

Quantisation

- Genetic algorithm
- Minimise bits
- Test equivalence

Verification

- Verify equivalence
- If not, generate counterexample
- Augment testset
- Repeat



CEG4N: Counterexample-Guided NN Quantisation (2)

Model	Verifier	r	Iter.	Bits	Status
seeds_10x1	ESBMC	0.01	2	3,4	Success
		0.03	13	18,12	Success
		0.05	3	6,4	Timeout
	NNEQUIV	0.01	2	3,4	Success
		0.03	2	4,3	Success
		0.05	2	4,4	Success
seeds_15x1	ESBMC	0.01	4	5,2	Success
		0.03	5	8,6	Timeout
		0.05	7	8,7	Timeout
	NNEQUIV	0.01	2	5,2	Success
		0.03	2	4,4	Success
		0.05	3	5,3	Success

See all results in Batista et al., IEEE TCAD Journal (2023)

- Few iterations are needed to converge to a safe quantisation!

Example 2: Quantised Neural Control

Collaboration with:

- University of Birmingham

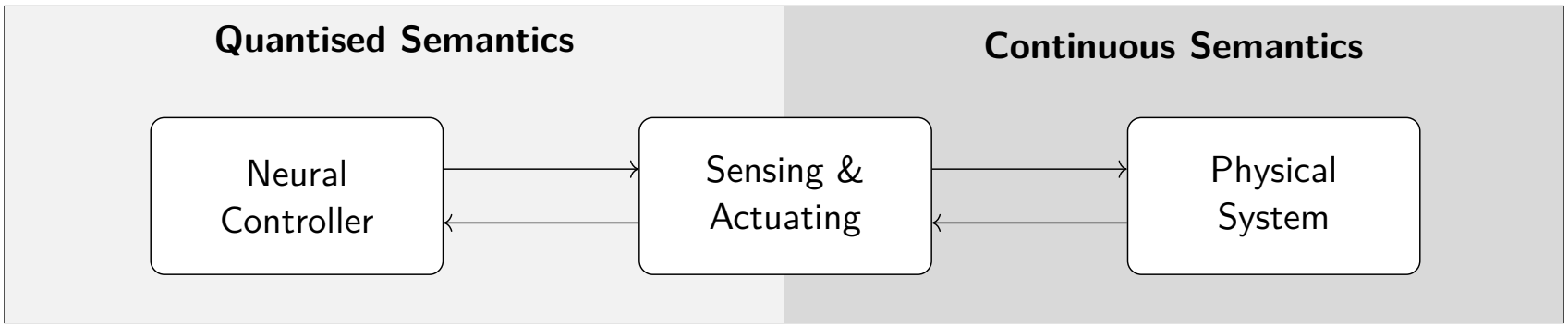
Funded by:

- Advanced Research + Invention Agency (ARIA)

See older position papers:

- Neural Network Verification is a Programming Language Challenge
- Floating-Point Neural Network Verification at the Software Level

Problem Setting



$$\hat{u} = \pi(\hat{x}), \qquad \hat{x} = \sigma(x),\ u = \alpha(\hat{u}), \qquad \dot{x} = f(x, u)$$

The goal is to synthesize a neural controller π, such that the system f is **guaranteed** to behave according to spec.

Safety and Liveness Specifications

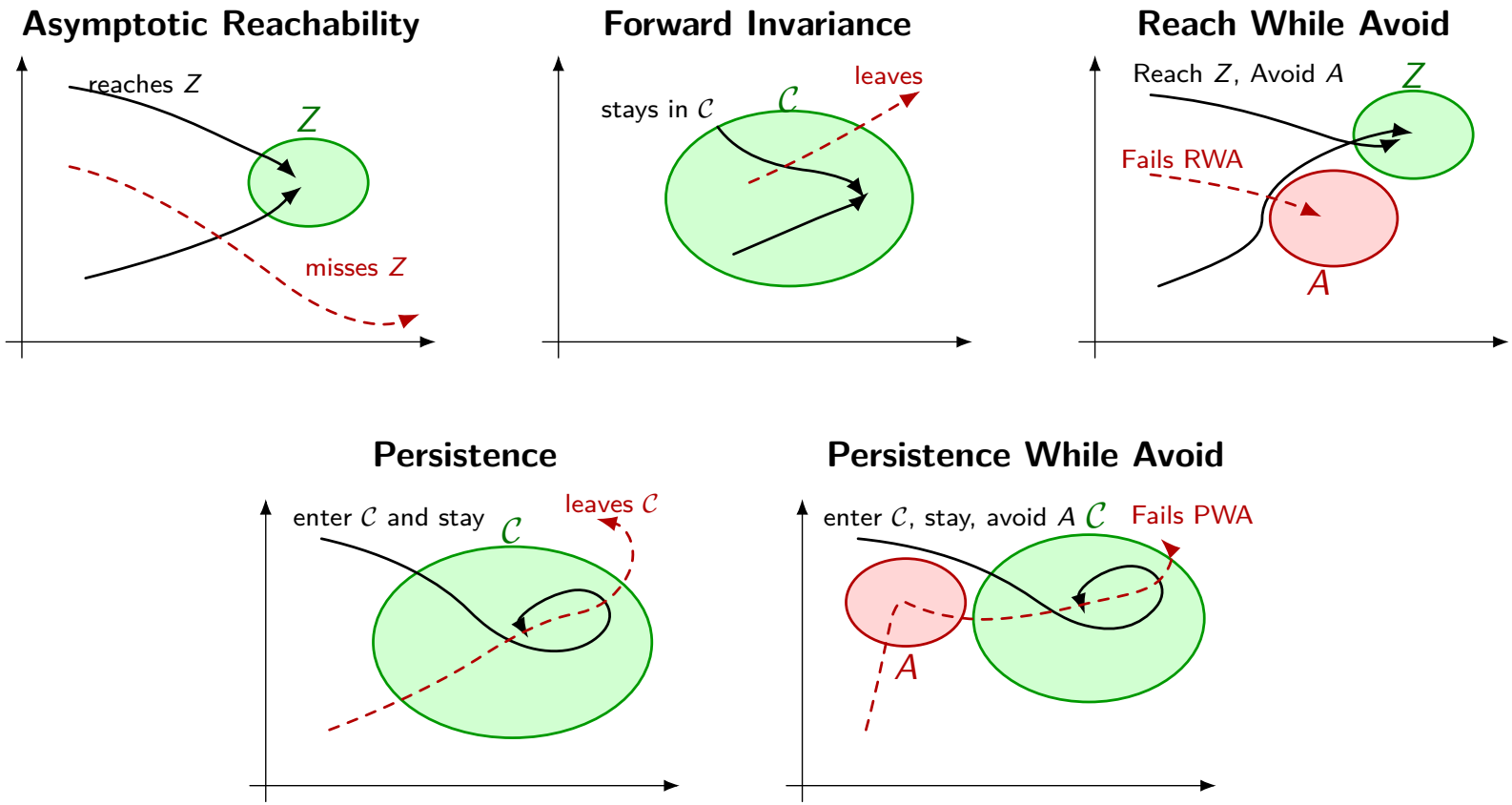
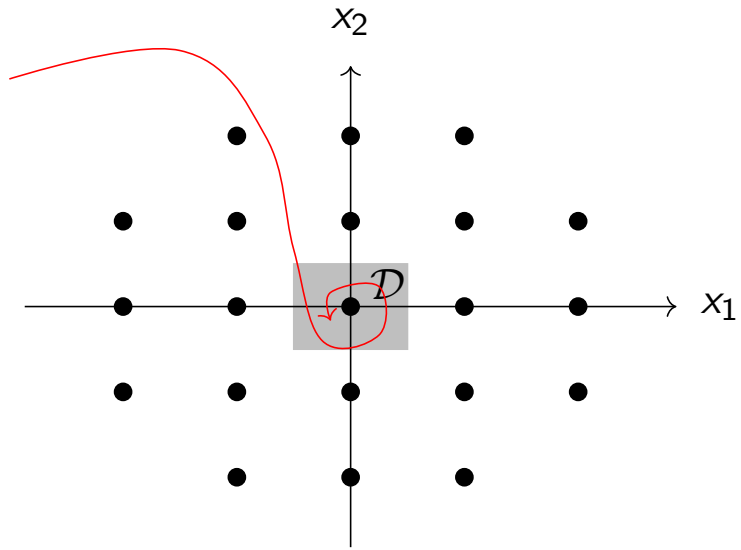


Table: Different properties of dynamical systems: reachability, invariance, reach-while-avoid, persistence, and persistence-while-avoid. Green sets are desired target/safe regions; red sets are avoid regions; dashed red trajectories illustrate violations.

Quantisation Issues

The "dead zone"

- Loss of control authority
- We may never converge



Non-uniform quantisation

- Even in integer arithm.
- $\pi(\hat{x})$ may have "gaps"

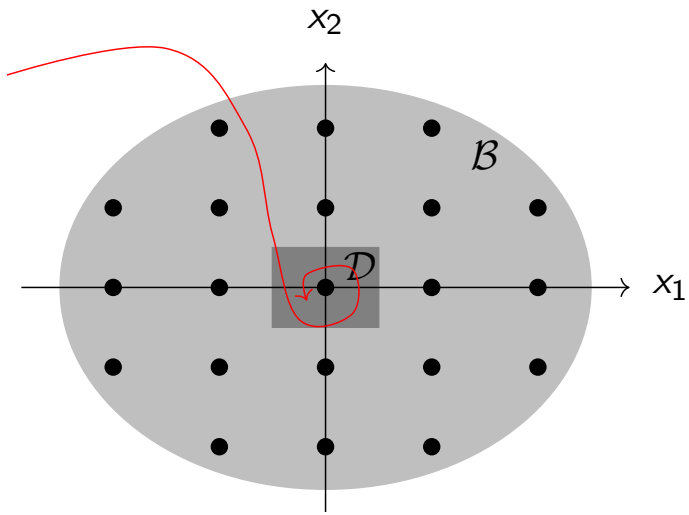
Goal:

- Work with neural controller in integer & floats.
- "Fix" existing theories based on Lyapunov certificates.

Lyapunov Certificates for Quantised Control

Stability certificate

- Find function $V : \mathbb{R}^d \to \mathbb{R}$
- such that $V(x) > 0$
- for all $x \in \mathcal{B}(0) \setminus \mathcal{D}\{0\}$
- in basin of attraction $\mathcal{B}$.



Then prove that:

$$\dot{V} = \nabla V \cdot f(x, \alpha(\pi(\sigma(x)))) < 0$$

arbitrary function (e.g. neural network)

neural controller (quantised)

physical system (continuous)

Example 3: Private Inference via FHE

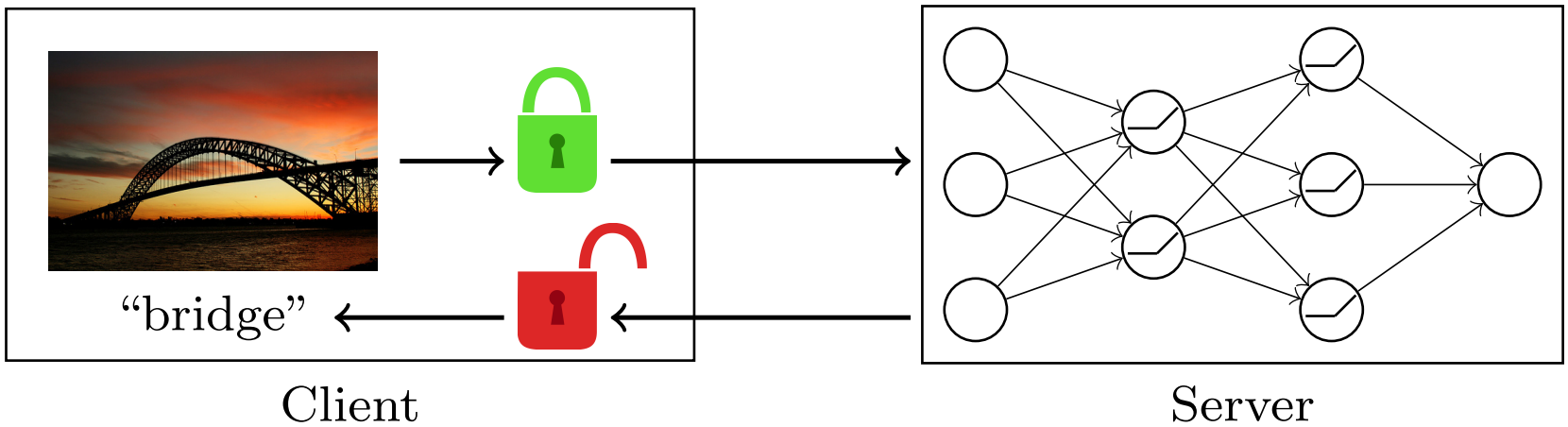
Ongoing collaboration with:

- Karlsruhe Institute of Technology
- Università degli Studi Milano-Bicocca

See papers:

- Certified Private Inference on Neural Networks via Lipschitz-Guided Abstraction Refinement
- Certified Error Analysis of Homomorphically Encrypted Neural Networks

Private inference for neural networks



An ideal model for machine learning as a service (MLaaS)

- The user sends out encrypted data
- The provider never sees the plaintext
- The user deciphers the NN output locally
- But how?

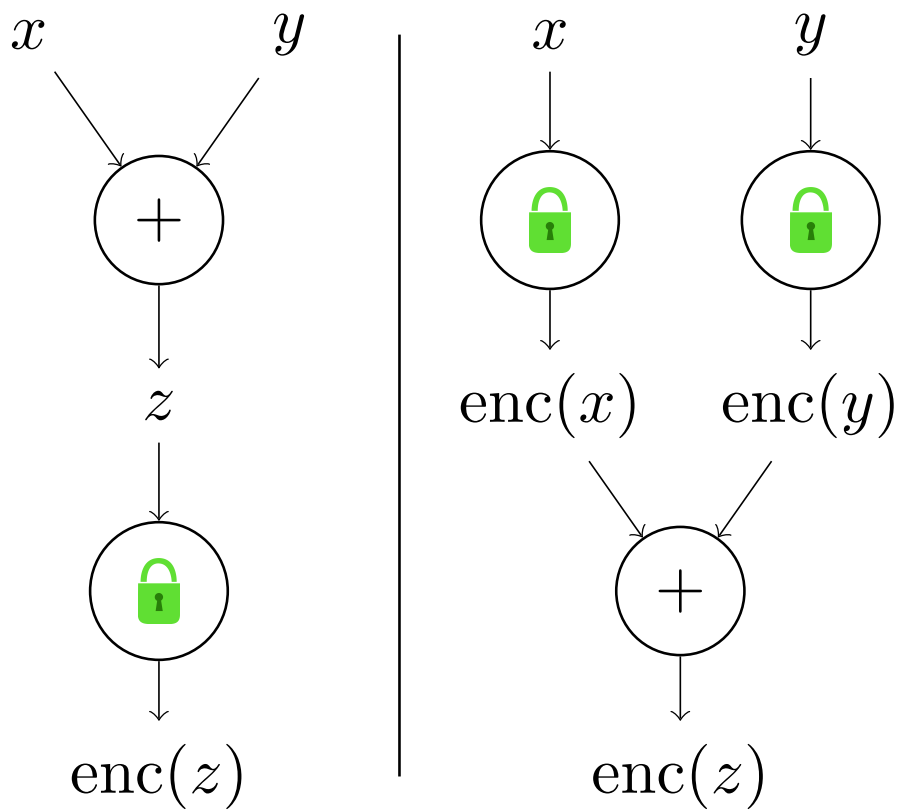
Fully-homomorphic encryption

Main idea

- $\text{enc}(x + y) =$
- $\text{enc}(x) + \text{enc}(y)$
- for all x, y

Best schemes

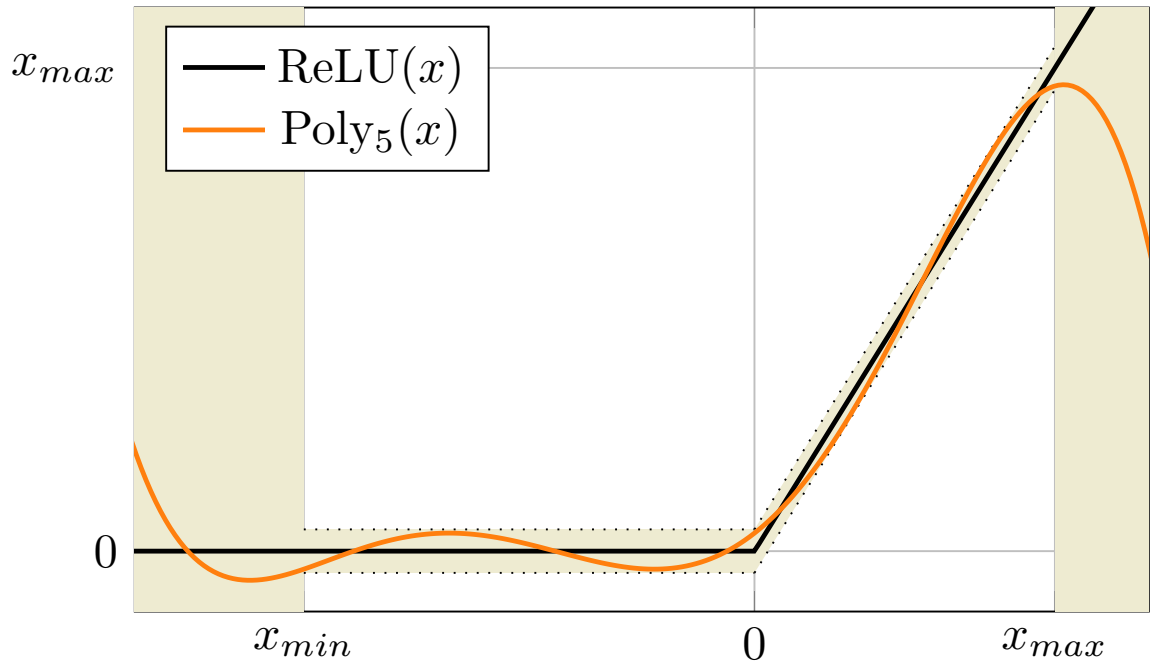
- Approximate (e.g. CKKS)
- Few operators
- Mainly $+$ and $*$



Application to neural networks is non-trivial

- The whole NN becomes a large polynomial!

Polynomial activations



General issues with polynomial activations

- The polynomial is "stable" in a limited range only
- Higher-degree polynomials are more expensive to compute

Existing private inference schemes

Retrain from scratch

- E.g. CryptoNets [2016]
- Uses x^2 activations
- Gradient instability

Approximate & fine-tune

- E.g. DeepReDuce, SNL
- Low-degree poly activations
- Escaping activation problem

Neural architecture search

- E.g. Delphi, SAFENet
- Keep a few ReLUs
- Replace the rest
- Requires garbled circuits

Post-training approximation

- E.g. Lee's work [2021-2022]
- High-degree poly activations
- **Equivalence problem**

Average case vs worst case

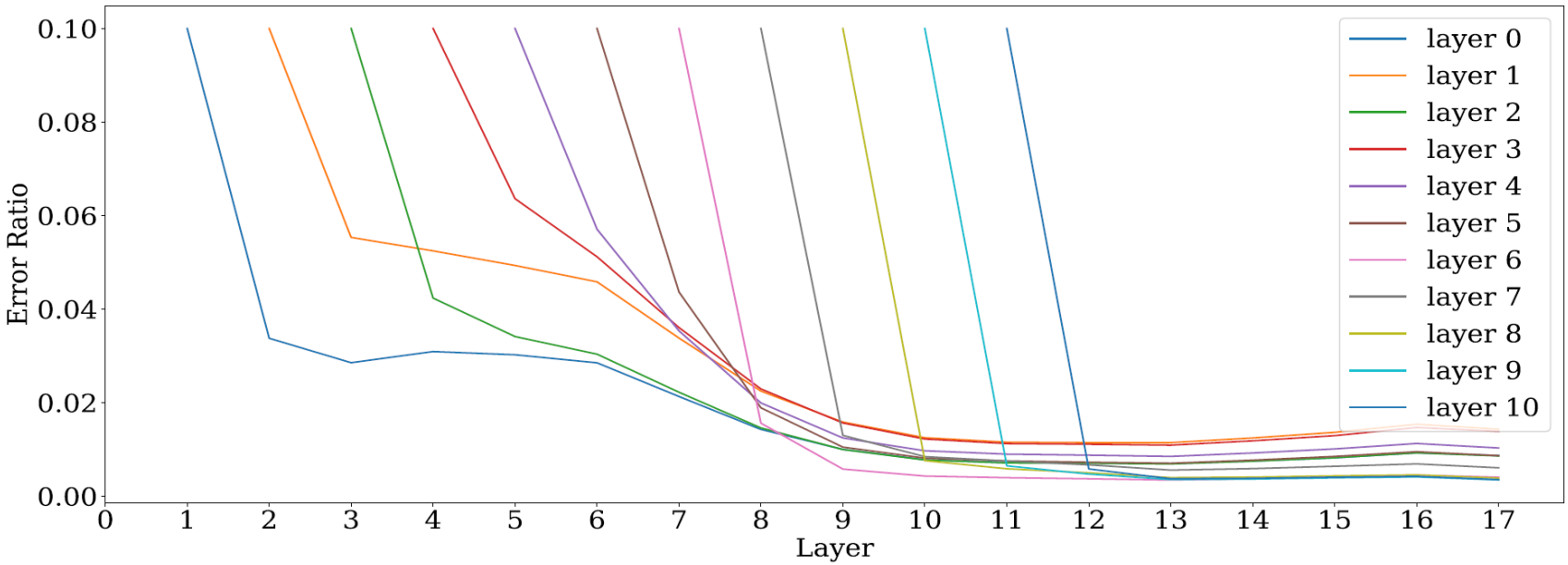
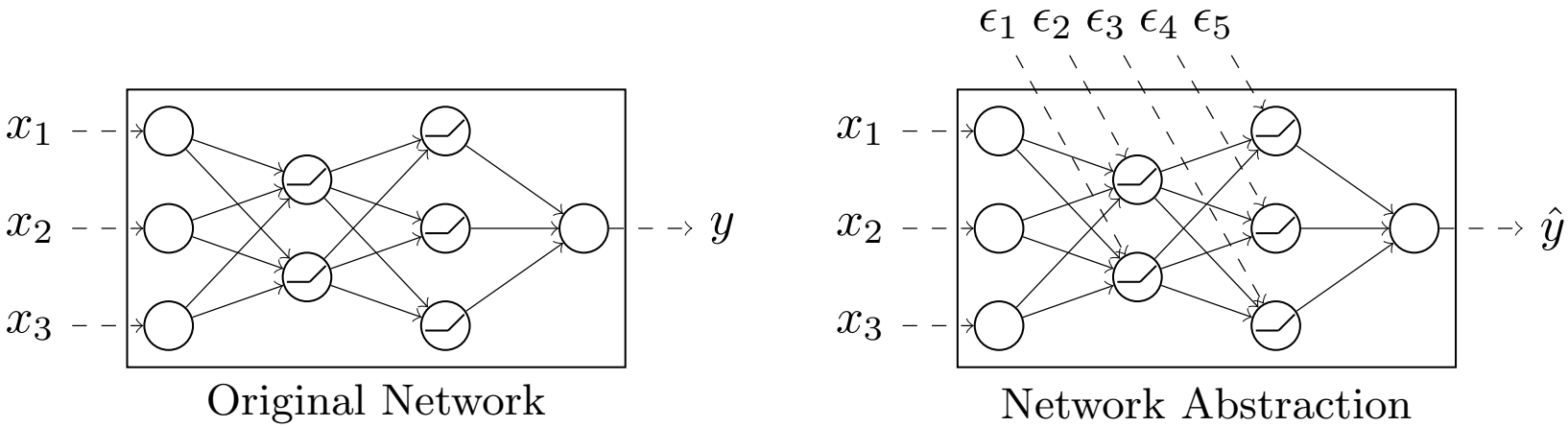


Figure: Attenuation of injected noise on a VGG-19 net trained on CIFAR-10. A curve starts at the layer where a scaled Gaussian noise is injected to its input (from Arora *et al.* ICML 2018).

Polynomial activations inject approximation error everywhere

- For most inputs x, the approximation errors cancel out
- However, we want to compute $\max_x |f(x) - \hat{f}(x)|$

Worst-case bounds error abstraction



Relaxation: sources of independent noise

- Introduces some level of over-approximation/slackness
- But allows for safe-by-design synthesis of FHE circuit

Better polynomial abstractions

- Extend the VeryDiff tool for tighter bounds
- Use smooth activations like GeLU instead of ReLU

Aerospace Standards

ED-324/ARP 6983

- Process Standard for Development and Certification/Approval of Aeronautical Safety-related Products implementing AI
- Requires evidence that the model semantics is preserved throughout the implementation phase

EUROCAE ED-76/RTCA DO-200

- Standards for Processing Aeronautical Data
- Controls the data flow from its creation to its use

They regulate the whole AI/ML infrastructure!

Aerospace Standards - Example Requirements

Unambiguous specification

- Capture the model's full "analytical/algorithmic syntax and semantics"

Enabling verifiability

- The final integrated MLC, running on the target hardware, must behave as specified

Guaranteeing traceability and reproducibility

- The model description is "completely developed" into the item-level implementation
- All development artifacts are identified, controlled, and retrievable

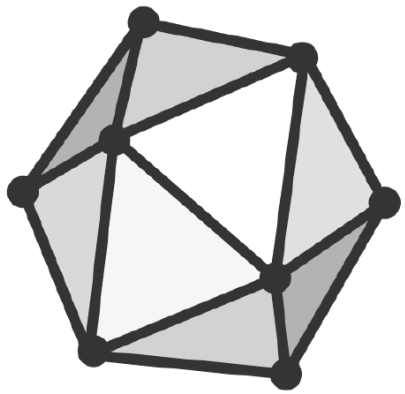
SONNX: Safety-Related ONNX Profile (1)

Focus on the Machine Learning Model Description (MLMD)

- Final output of the model design phase
- Primary input specification for the implementation phase

Formalisation of existing ONNX standard

- Open Neural Network eXchange format
- Make it unambiguous, rigorous, verifiable, interoperable

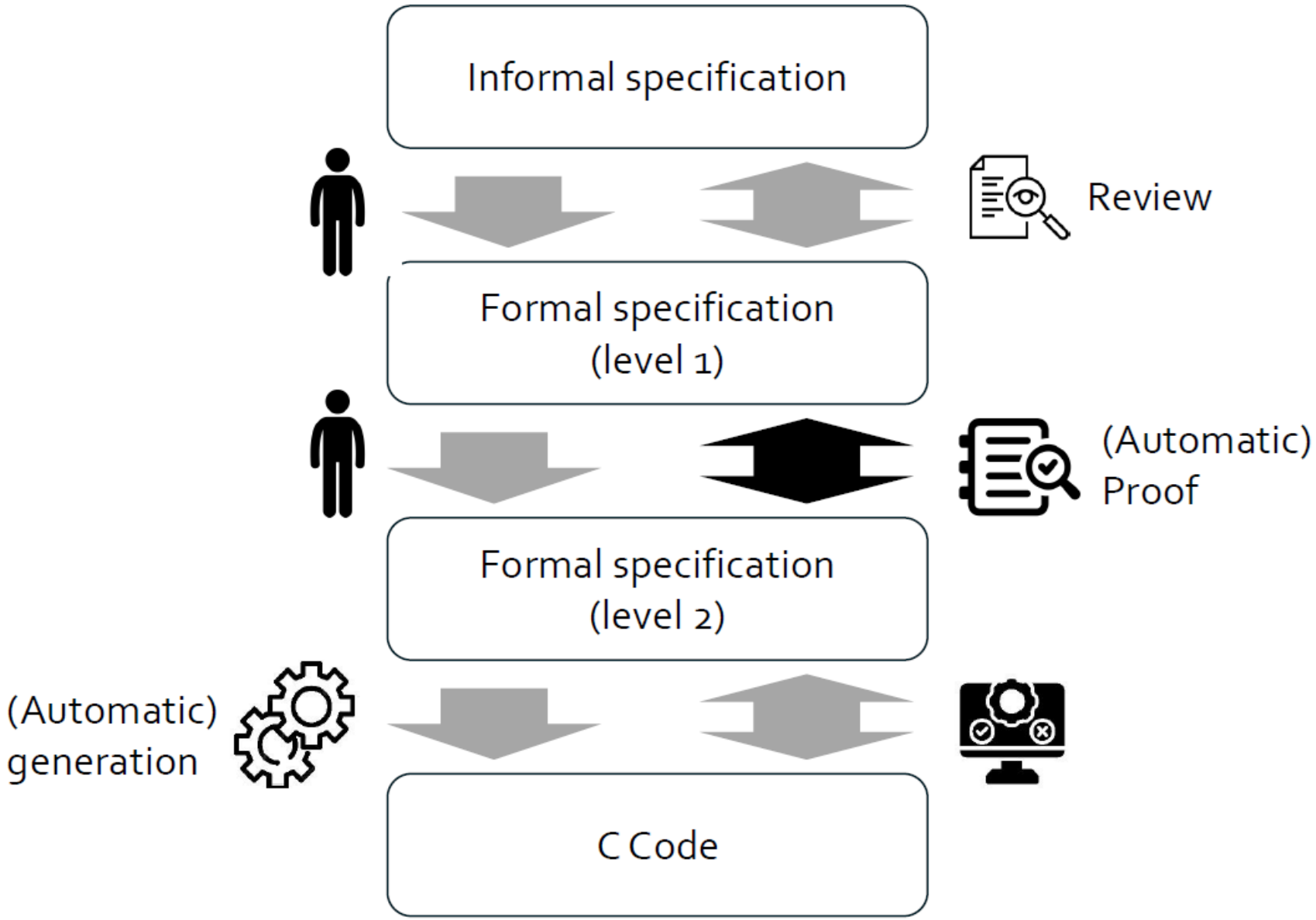


Supported by

- Airbus, Thales, Bosch, Collins, Linux Foundation, ...

See `https://github.com/ericjenn/working-groups/tree/ericjenn-srpwg-wg1/safety-related-profile`

SONNX: Safety-Related ONNX Profile (2)



SONNX: an Open Challenge

SONNX reference impl.

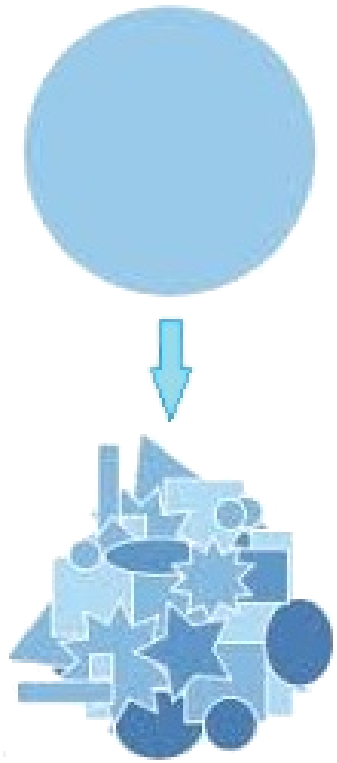
- Automatically generates C code
- Very naïve implementation
- E.g. single thread CPU

Modern ML frameworks

- Can use parallel hardware (GPUs)
- But they introduce non-determinism

Challenge: equivalence checking

- Build efficient, optimised implementations
- with $\leq$ numerical error than the naïve one



Summary

Neural network verification exists!

- And it scales to models with 100M+ params
- Full tutorial next week at AAAI 2026

Applications to SW/HW domain

- Safe quantisation and compression
- Safe control and robotics
- Privacy-preserving inference

ML infrastructure challenges

- Compliance with ED-324/ARP 6983
- Safe and reproducible exchange formats

Any questions?