Neural Network Verification for Certified Private Inference

Edoardo Manino

Lecturer in AI Security

27 May 2026

Table of Contents

NN Verification Intro: Adversarial Attacks

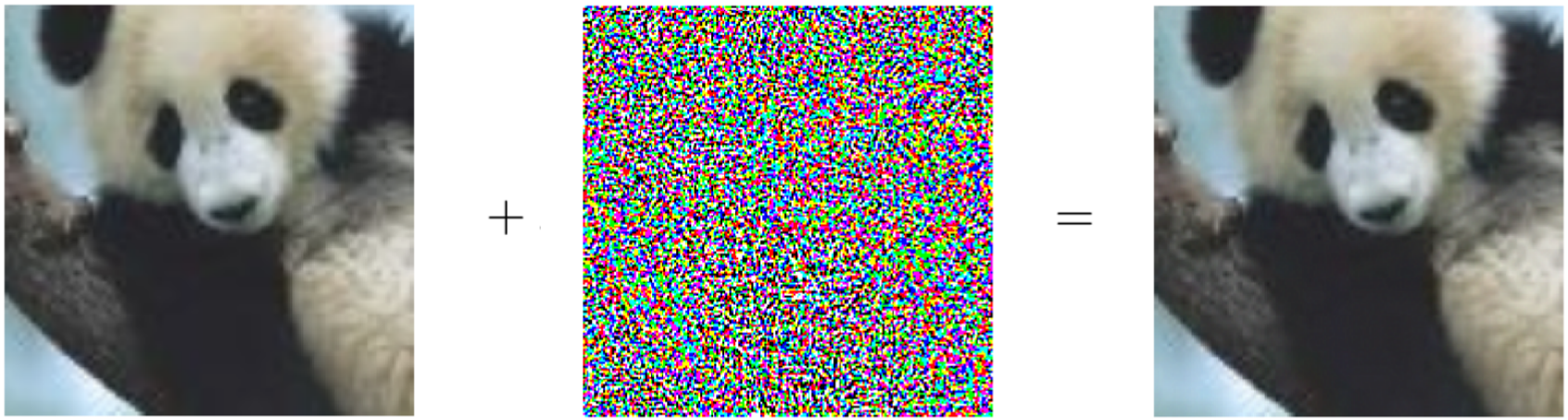
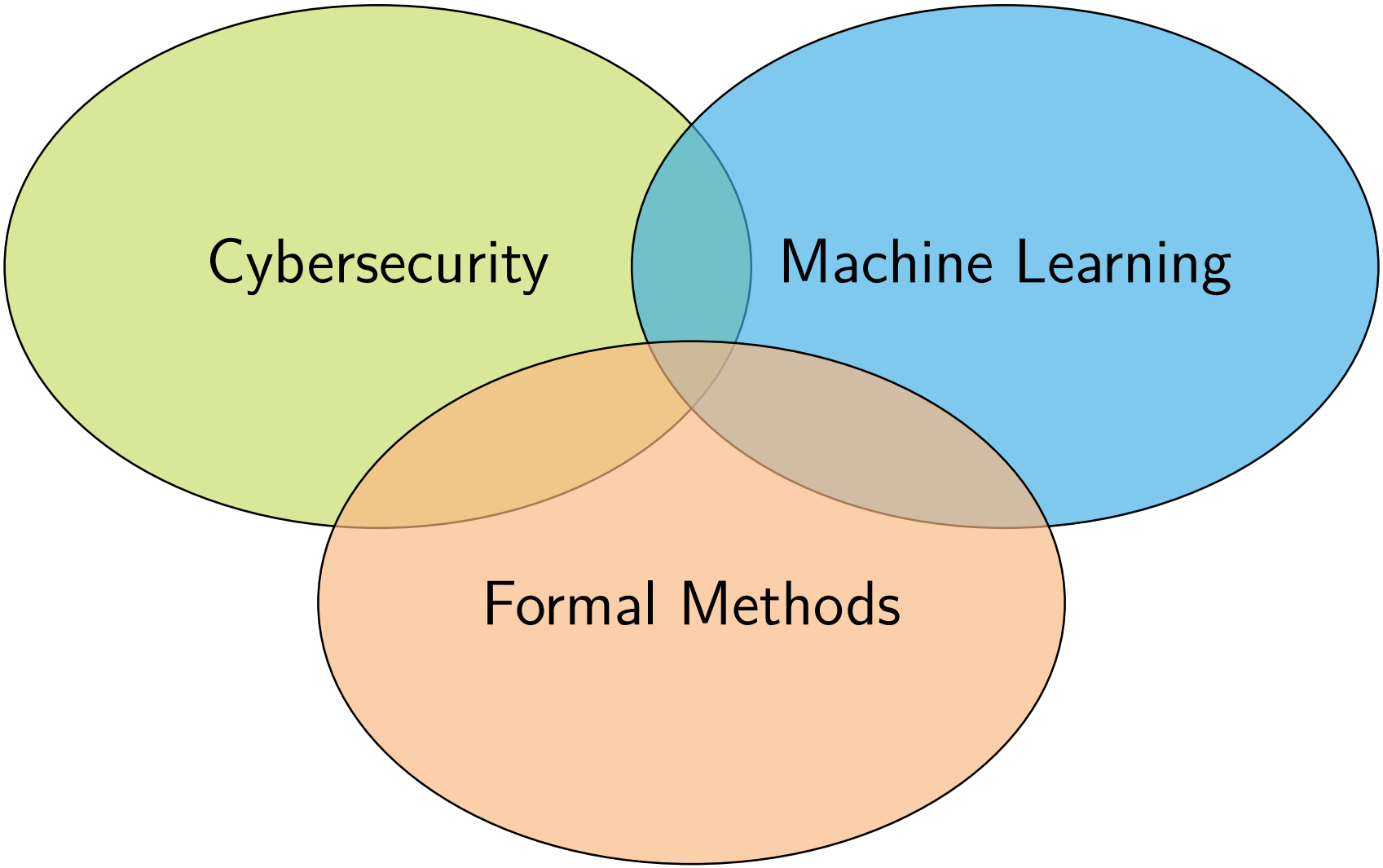


Figure: Image adapted from Goodfellow *et al.*, 2015.

Neural networks are *not* robust to input noise

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

NN Verification Intro: Fertile Ground for Research



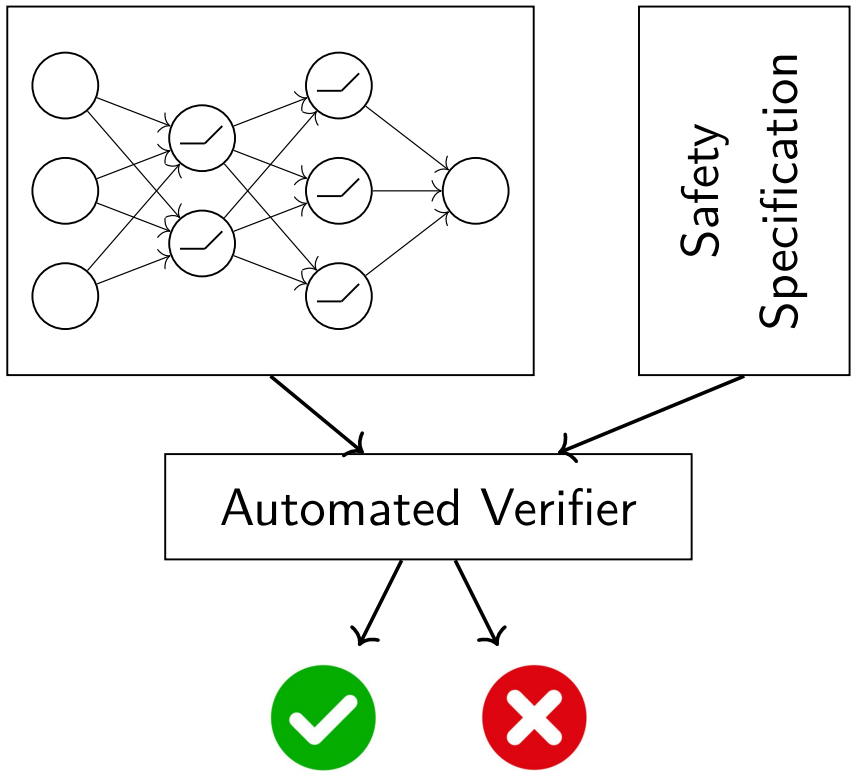
NN Verification Intro: High-Level Definition

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

NN Verification Intro: Specifications

Neural network as a composition of L layers

- $x_0 = x, \qquad x_\ell = \sigma(W_\ell x_{\ell-1} + b_\ell), \qquad f(x) = x_L$

Specifications as pre- and post-conditions

- $\forall x \in \mathcal{I}n, \quad f(x) \in \mathcal{O}ut$

Adversarial robustness (again)

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

Other classic properties

- Monotonicity, Equivalence $\rightarrow$ product of inputs or networks
- Stability, Persistence, Avoidance $\rightarrow$ more on these later

NN Verification Intro: Algorithm Overview

Algorithms combine

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

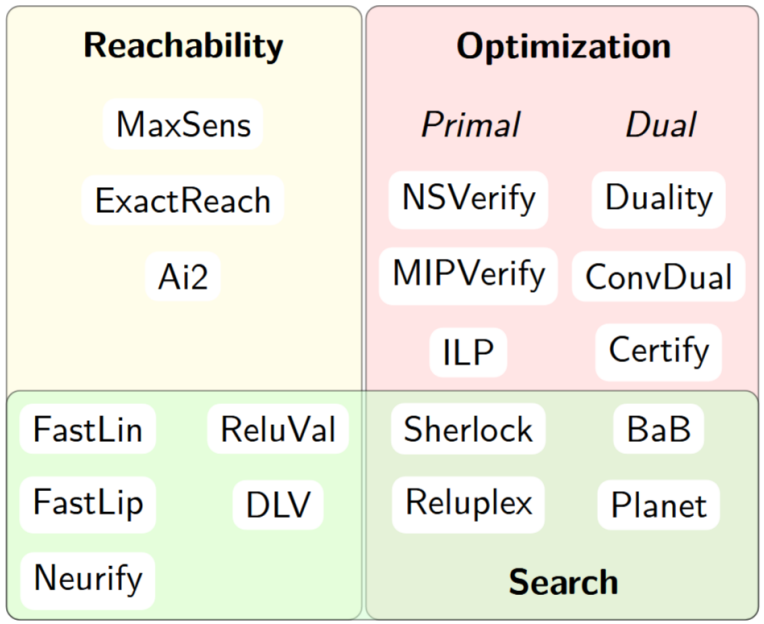
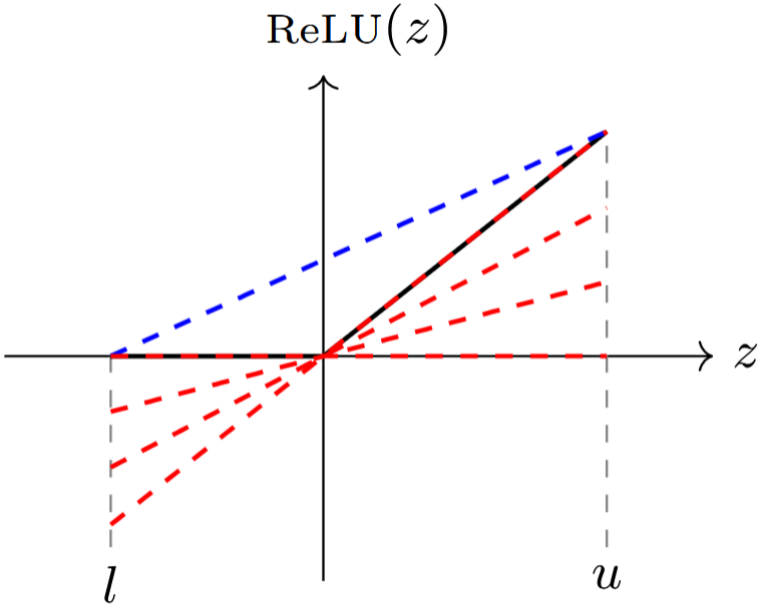


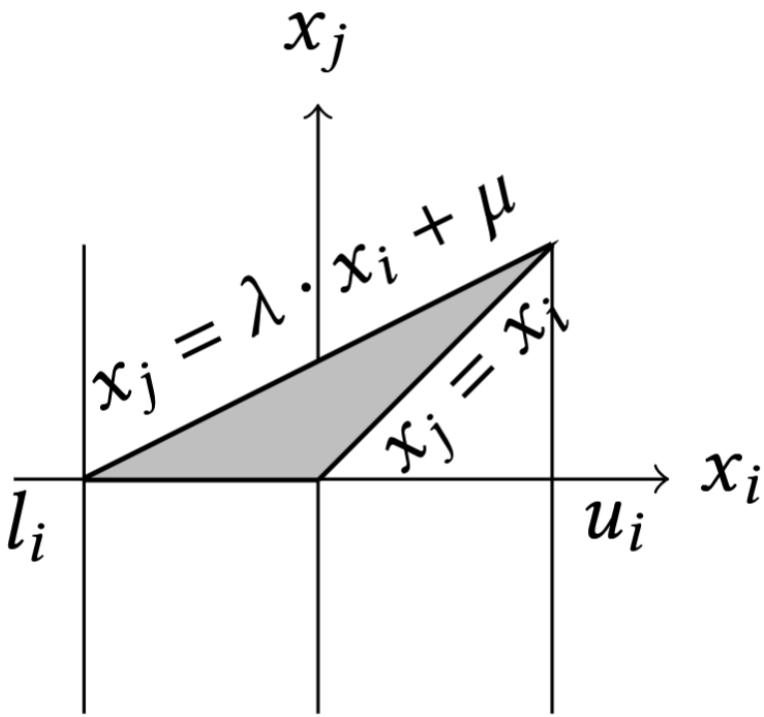
Figure: Image by Liu *et al.*, 2021

Remaining challenge

- Find the right combination to scale to (very) large networks!

NN Verification Intro: Dealing with ReLUs





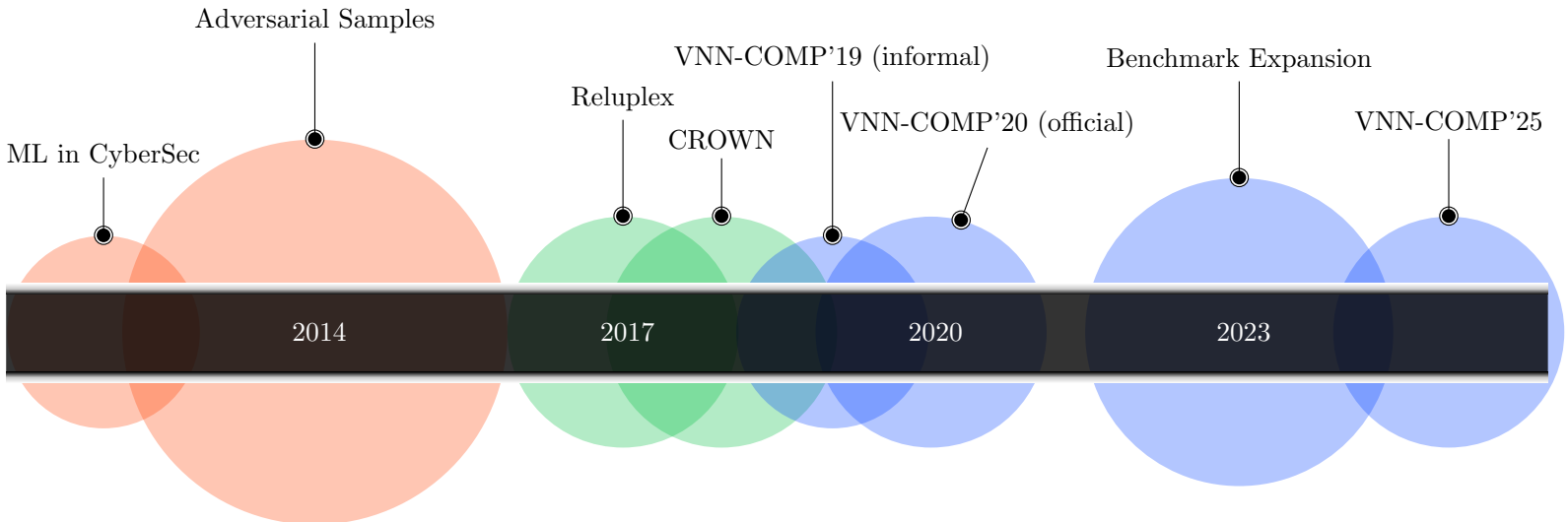
The α-CROWN strategy

- Parametric linearisation
- Can adjust lower bound
- End-to-end optimisation

The DeepPoly strategy

- Linear constraints
- Three per each ReLU
- Solve LP problem

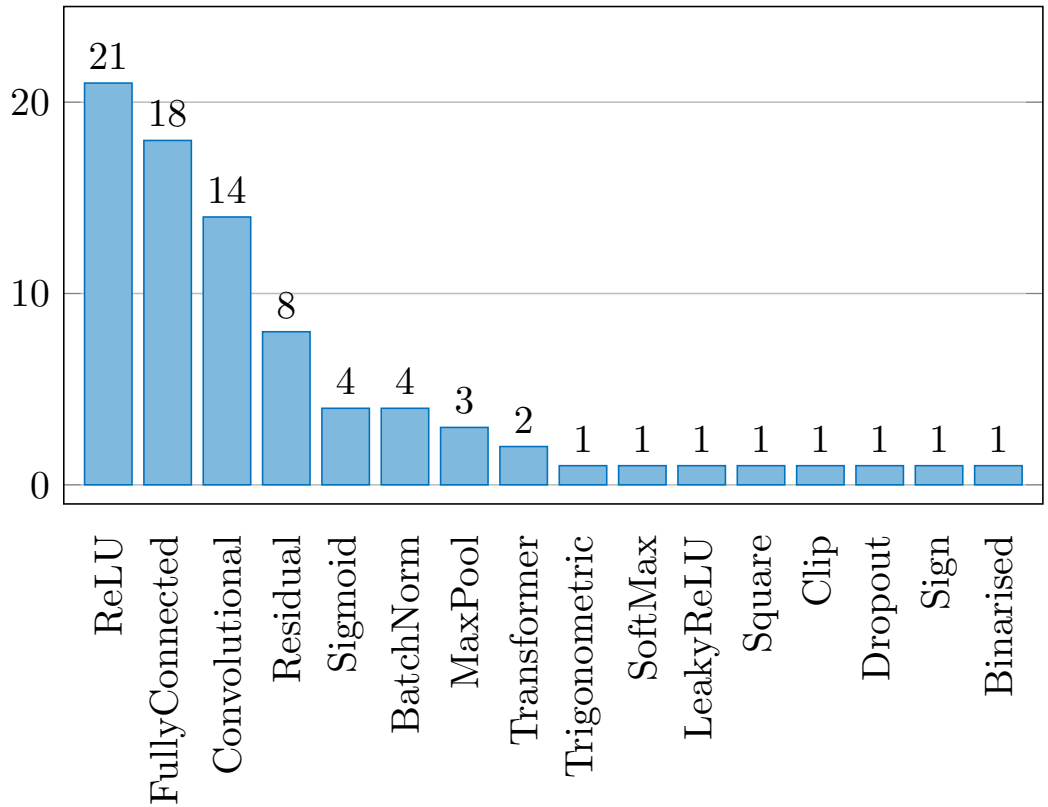
NN Verification Intro: a Timeline

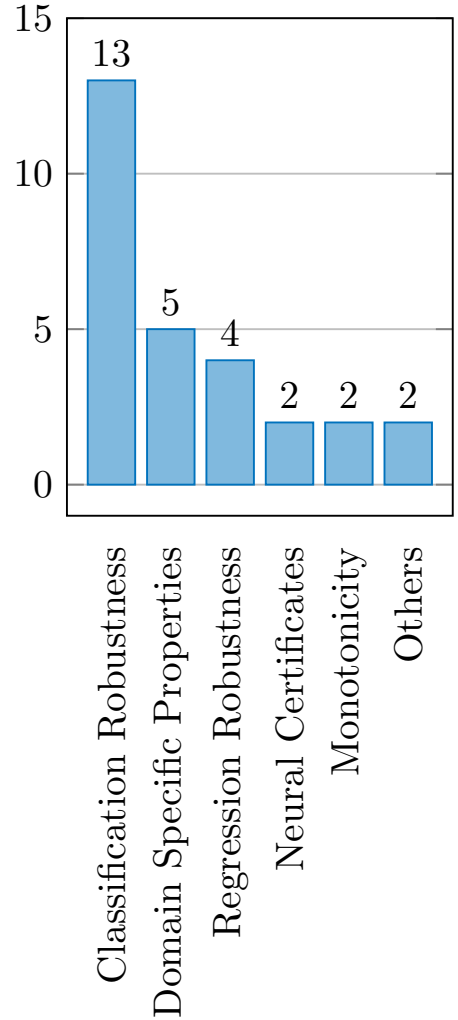


A maturing field

- Well-established tools: e.g. $\alpha\beta$-CROWN, Marabou, PyRAT
- VNN-COMP international competition (2026 edition ongoing)
- Standard input formats: ONNX + VNN-LIB 1.0 & 2.0

NN Verification Intro: Layer Types and Use Cases





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

NN Verification Intro: Scalability Demonstration

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 @ 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

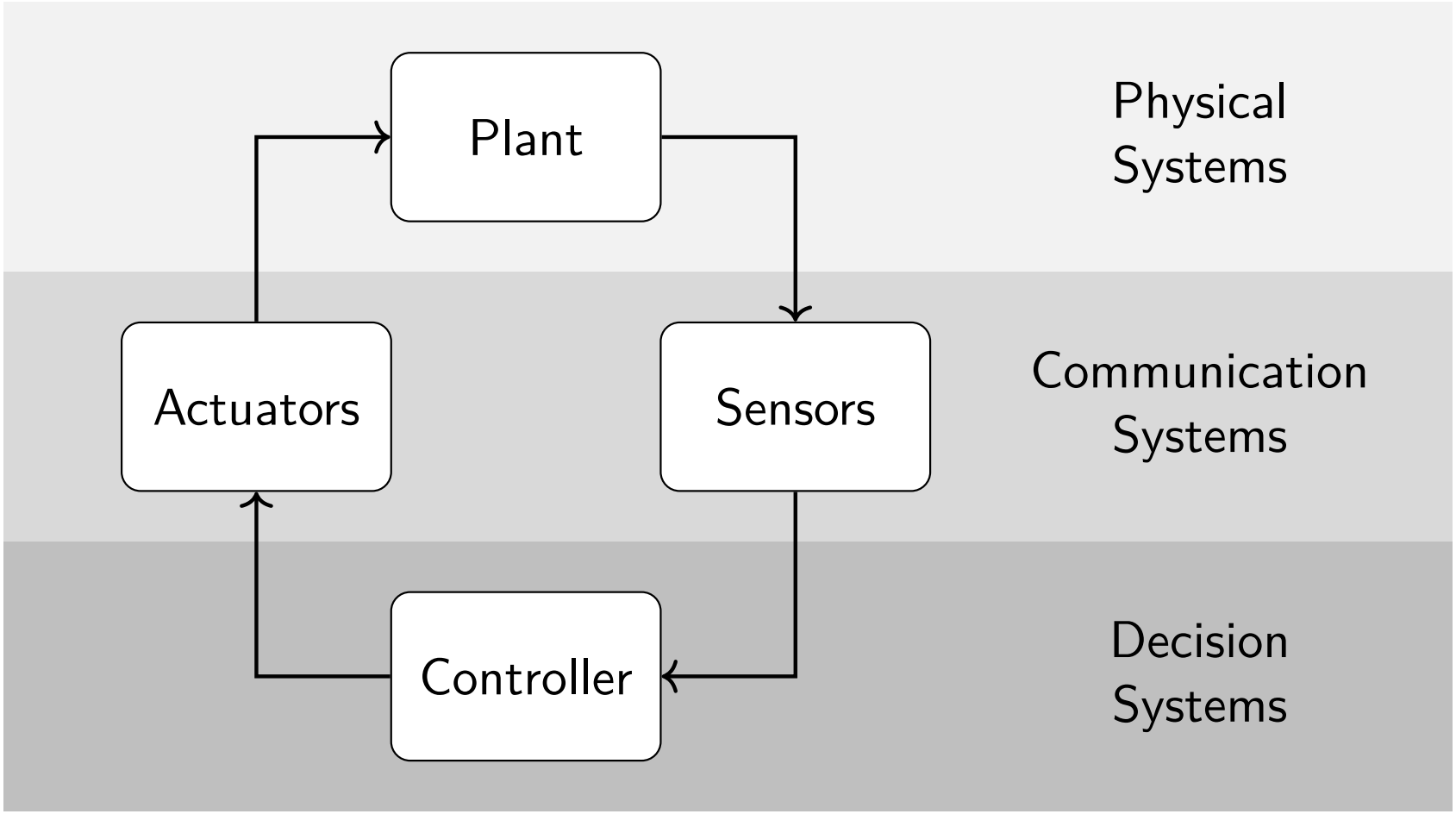
AAAI Tutorial & Lab Forum | Official Lab Listing

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

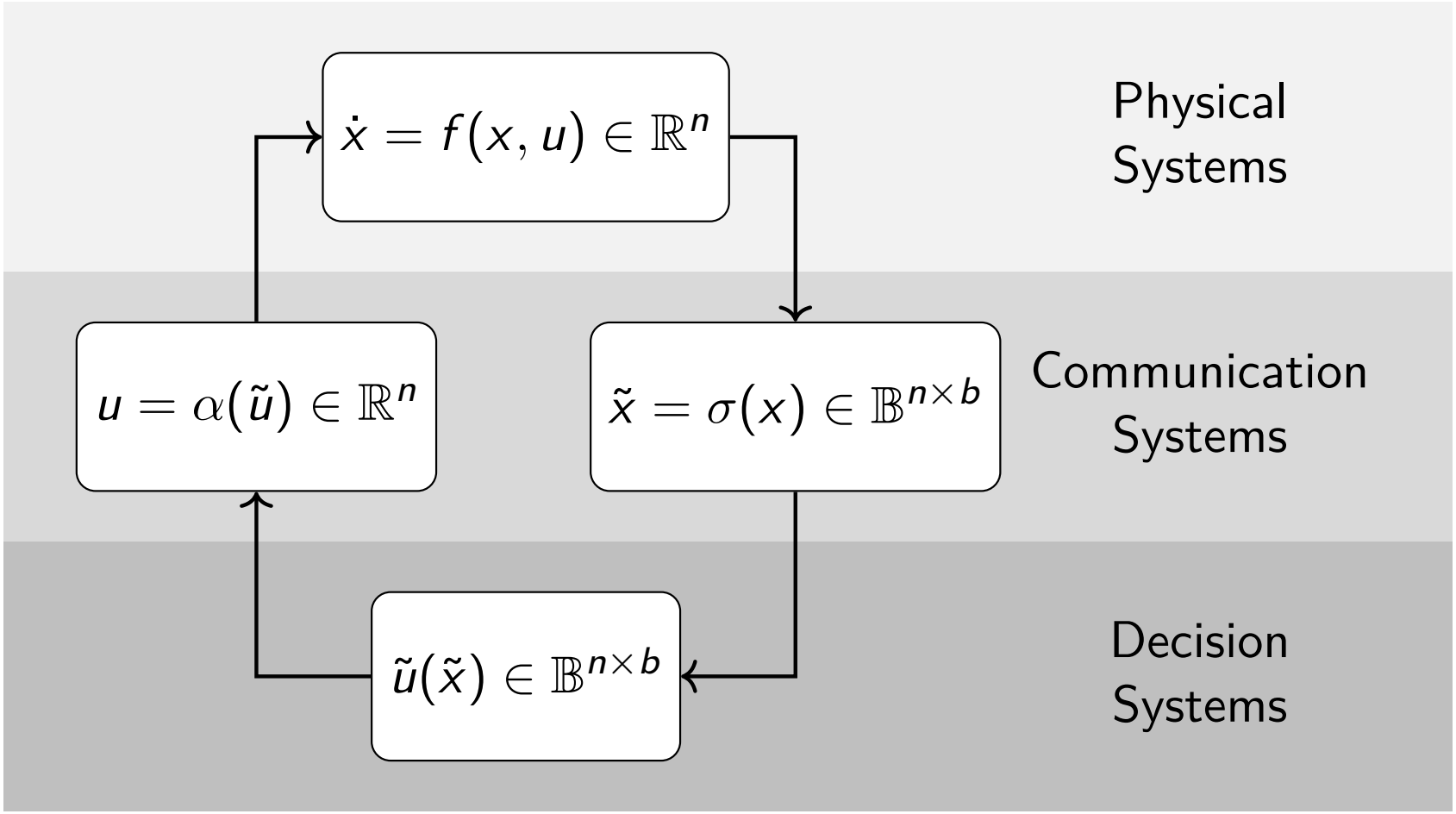
With additional contributions from Stanley Bak, Tobias Ladner, and other VNN-COMP organizers

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

Safe Control: a Closed-Loop Problem



Safe Control: Semantic Mismatches



Safe Control: Closed-Loop Specifications

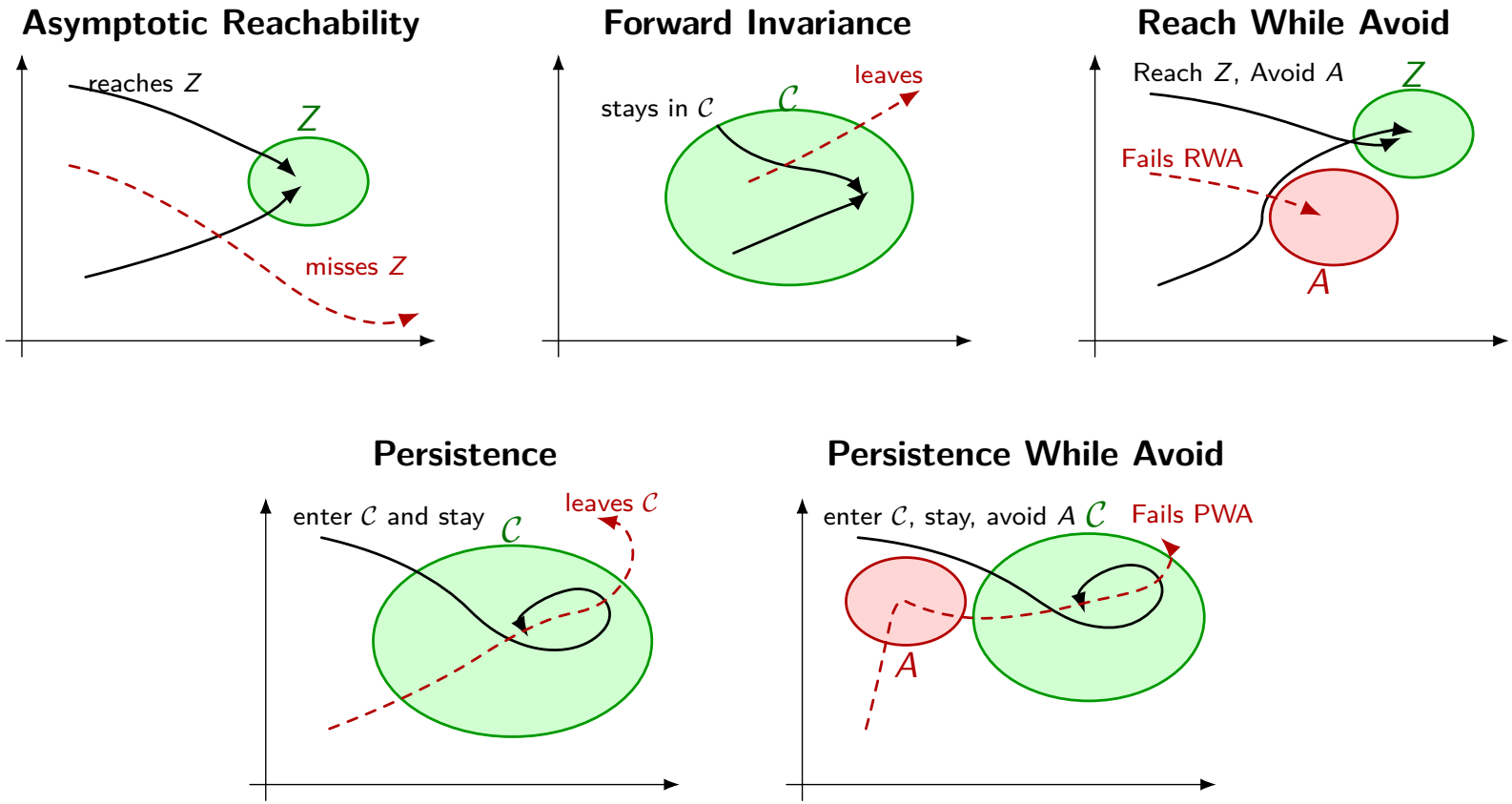
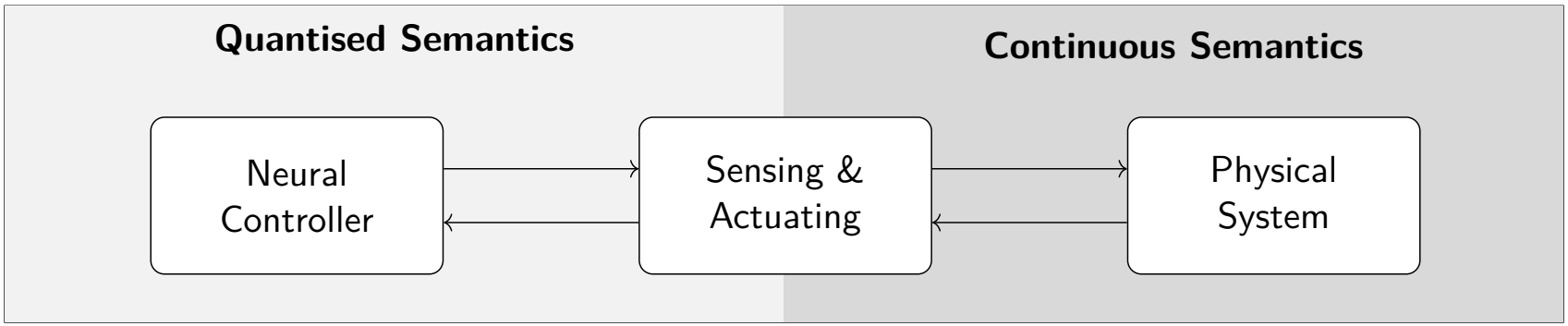


Table: Different properties of dynamical systems: reachability, invariance, reach-while-avoid, persistence, and persistence-while-avoid. Green sets are desired target/safe regions; red sets are avoid regions.

Safe Control: (Quantised) Neural Controllers

In collaboration with:

- University of Birmingham
- Advanced Research + Invention Agency (ARIA)



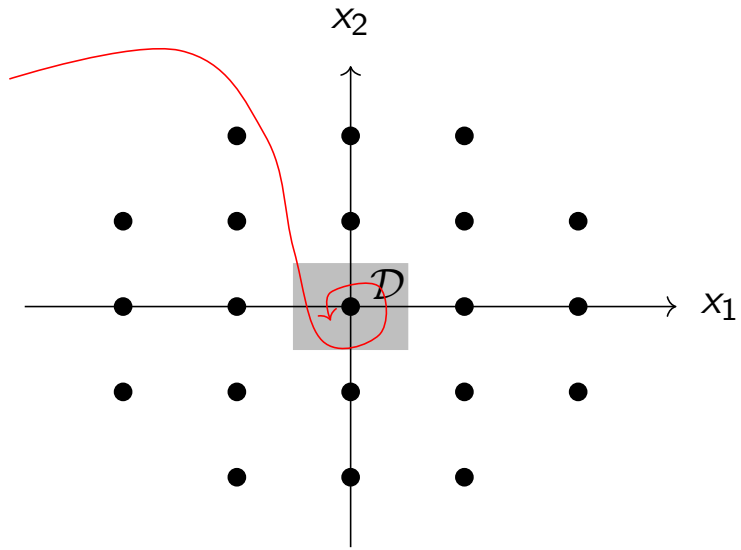
See older position papers:

- Neural Network Verification is a Programming Language Challenge, ESOP 2025
- Floating-Point Neural Network Verification at the Software Level, arXiv 2025

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"

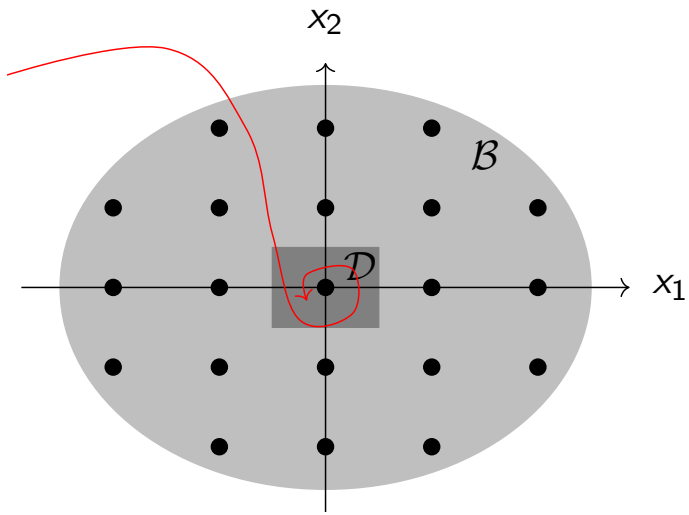
Goal:

- 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)

Certified Private Inference

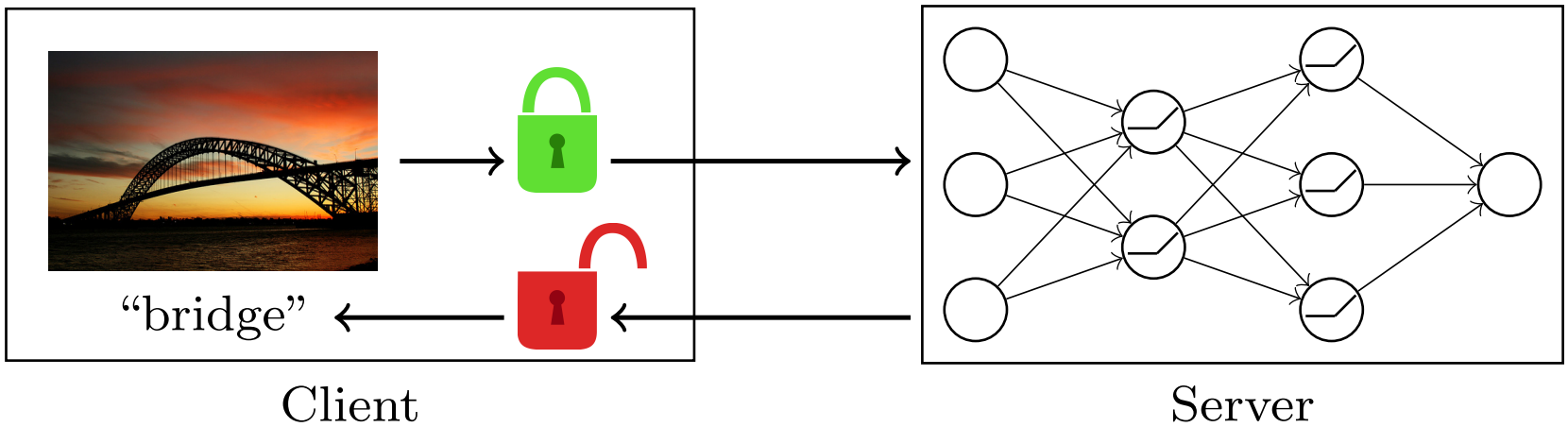
Ongoing collaboration with:

- Karlsruhe Institute of Technology
- Università degli Studi Milano-Bicocca
- Politecnico di Torino

See papers:

- Certified Private Inference on Neural Networks via Lipschitz-Guided Abstraction Refinement, FoMLAS 2023
- Certified Error Analysis of Homomorphically Encrypted Neural Networks, SAIV 2025
- Encrypted Neural Networks without Overflows, arXiv 2026

Certified Private Inference: Problem Setting



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?

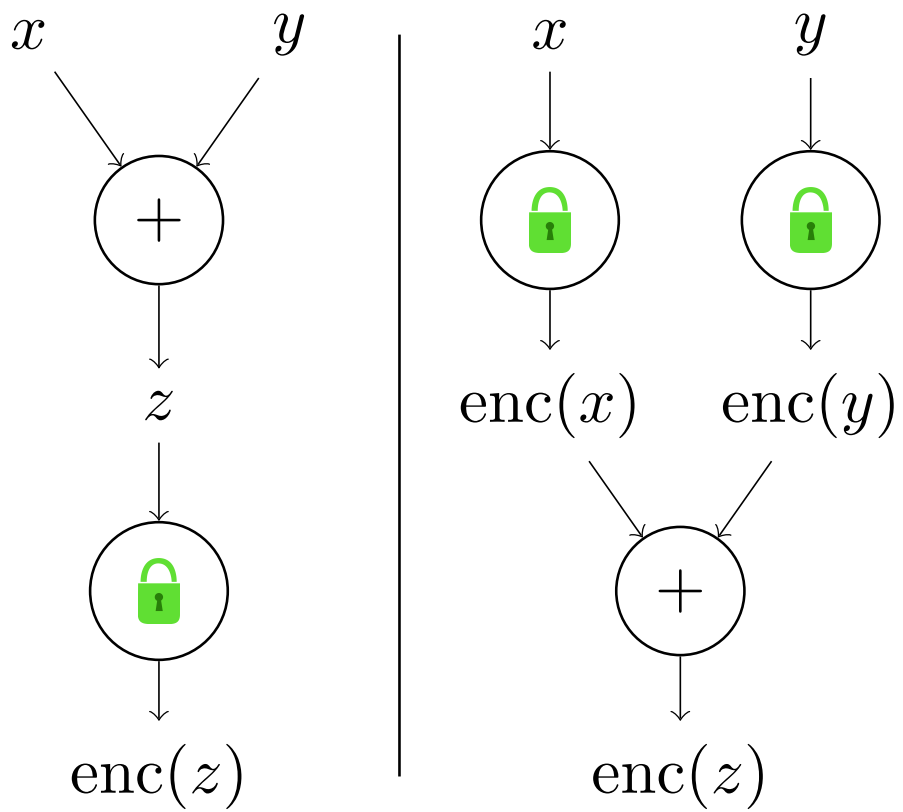
Certified Private Inference: Homomorphic Encryption

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!

Certified Private Inference: Existing 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

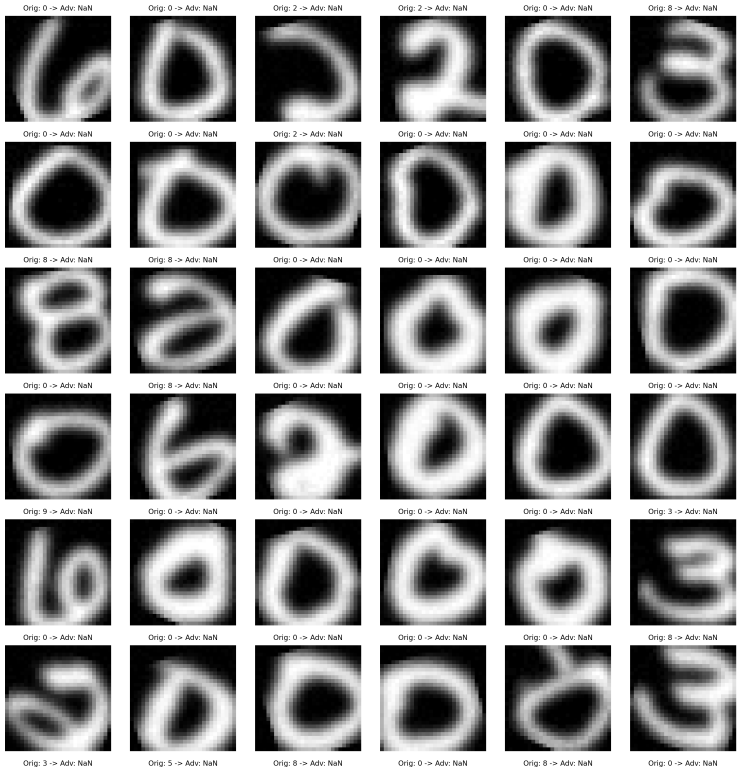
- E.g. Delphi, SAFENet
- Keep a few ReLUs
- Replace the rest
- Requires garbled circuits

Post-training approximation

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

Latest results [2024-2026]: scaling to 8B parameters LLMs

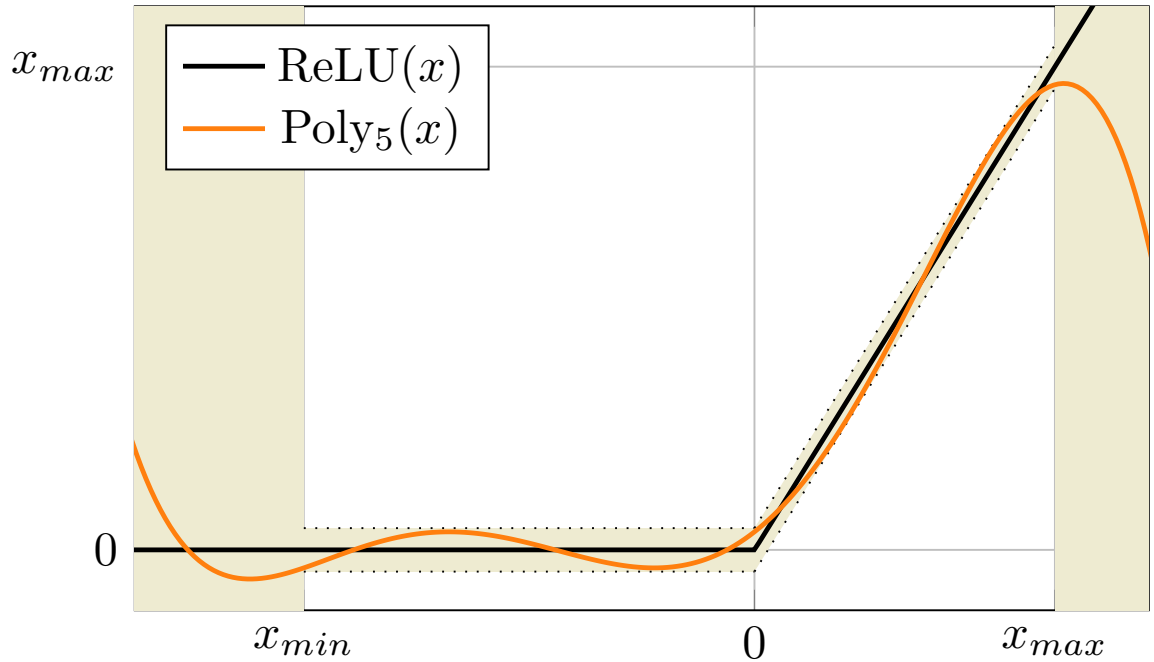
Certified Private Inference: Overflow Attacks



Our find: FHE neural networks are not reliable

- Legitimate inputs can cause overflows in the encrypted circuit
- The cyphertext becomes corrupted and cannot be decrypted

Certified Private Inference: Polynomial Activations



General issues with polynomial activations

- The polynomial is "stable" in a limited range only
- The approximation error can accumulate through the network

Certified Private Inference: Sampling is not Enough

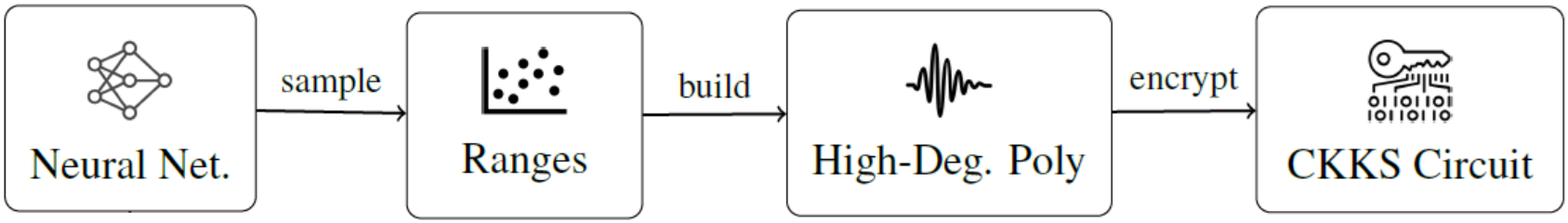
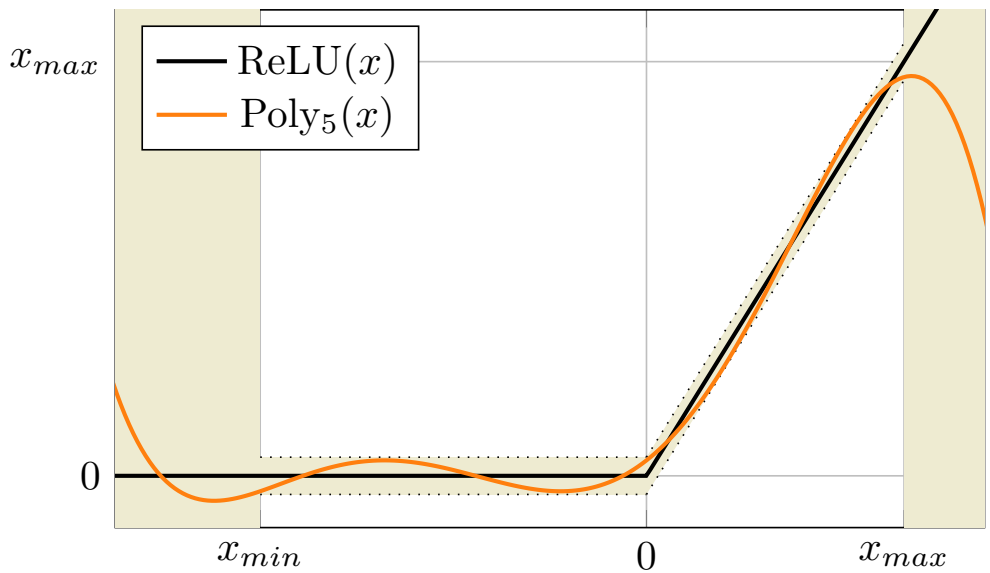


Figure: Modern workflow for designing FHE neural networks



Certified Private Inference: Sampling is not Enough

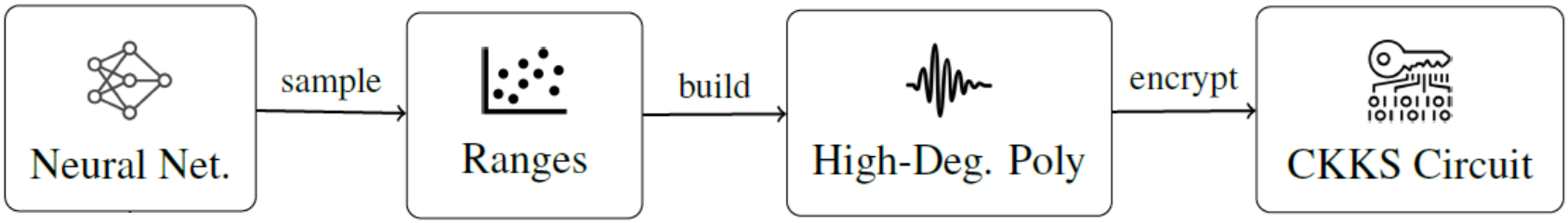


Figure: Modern workflow for designing FHE neural networks

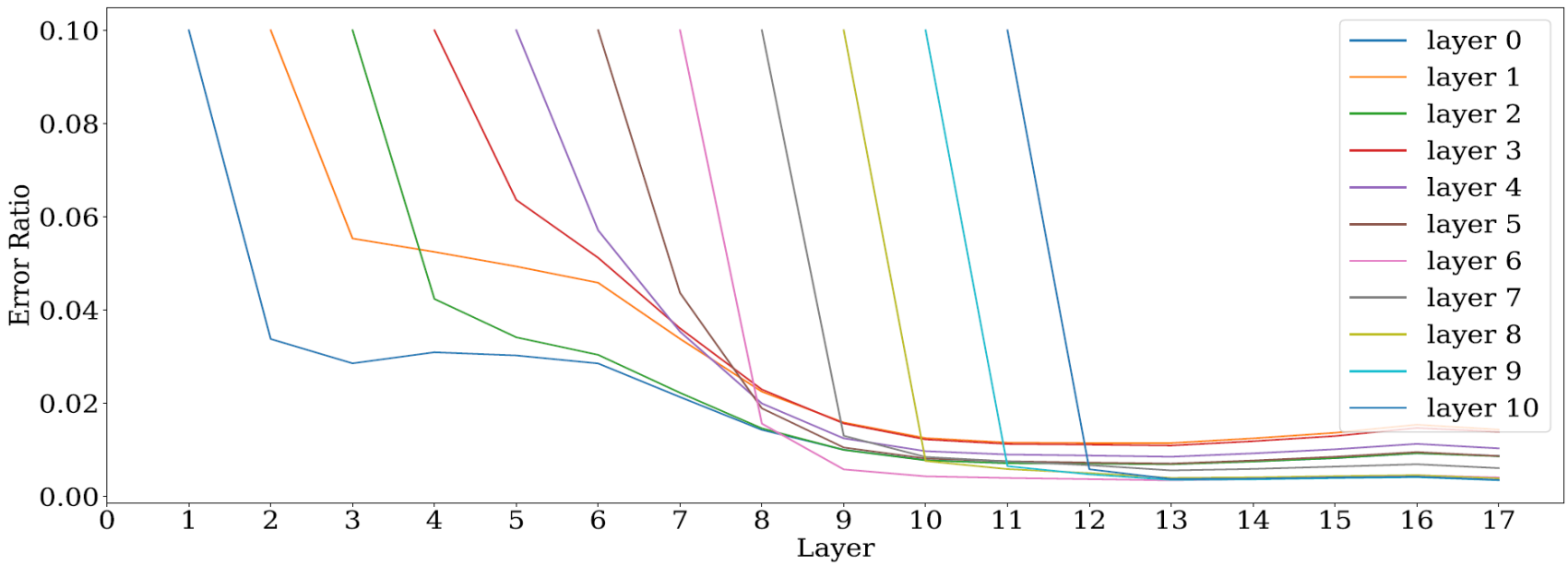
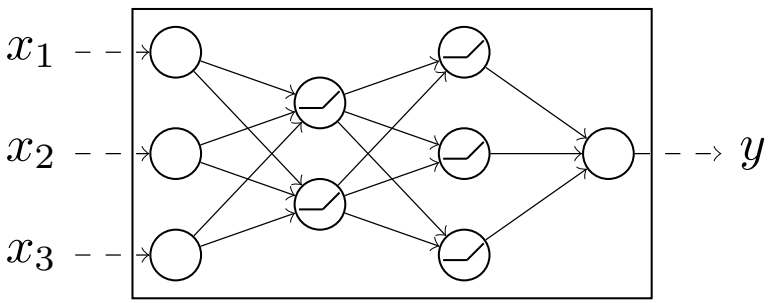
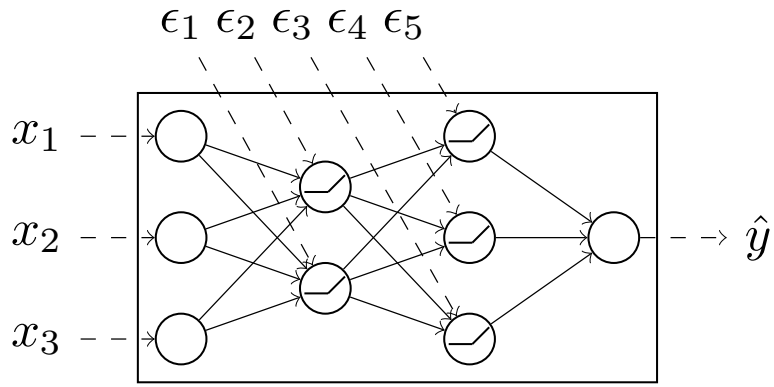


Figure: Attenuation of injected noise on a CIFAR-10 net (Arora 2018).

Certified Private Inference: Worst-case Abstraction



Original Network



Network Abstraction

Abstraction: add sources of independent noise

- Bound the worst-case error at every activation
- $\forall x \in [0,1]^d, \forall \epsilon_{<i} \in [-\delta_{<i}, \delta_{<i}], \implies |p_i(x) - \mathsf{ReLU}_i(x)| \leq \delta_i$

It is a differential verification query!

- We want to prove equivalence between $f(x)$ and $\pi(x)$
- Up to some design approximation error δ

Certified Private Inference: Overflow-Free Design

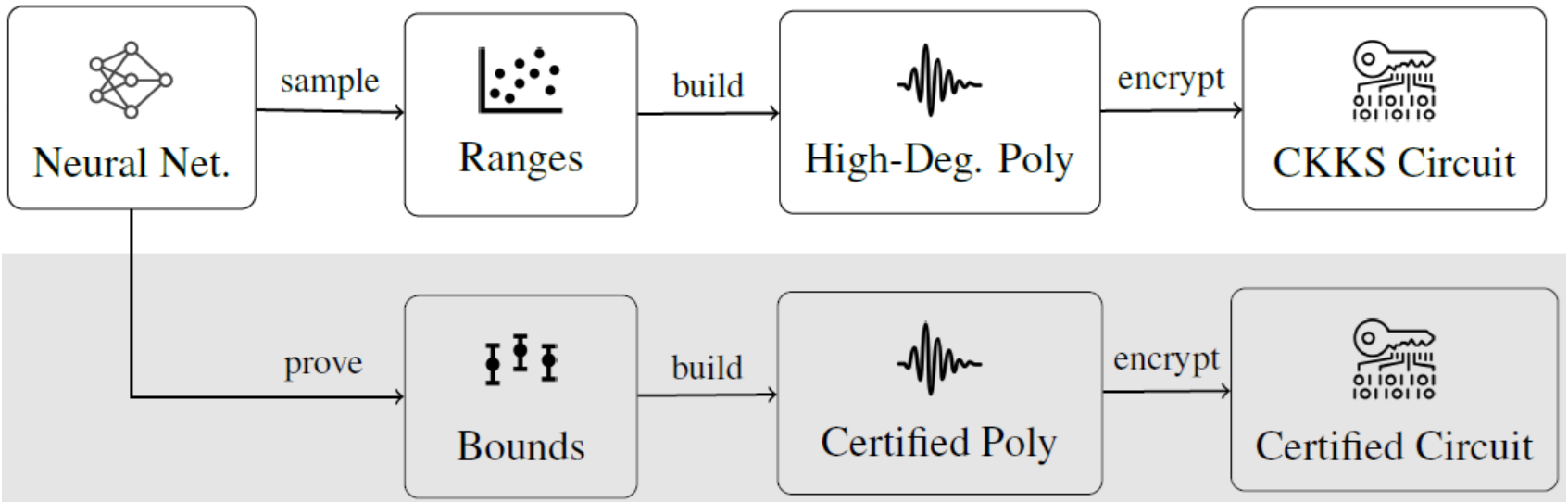


Figure: Modern workflow vs certified workflow for FHE neural networks

Our certified design workflow

- Computes guaranteed bounds on the activation ranges
- Designs tight polynomial approximations across them
- Thus ensuring no overflows or error explosion

Certified Private Inference: Output Error

Network	Size	Accuracy		Attack Success (smaller is better)	
		Sampled	Certified	Sampled	Certified
HELOC	[64, 32]	71.86%	71.86%	0.05%	**0.0%**
MNIST	4x256	97.80%	97.80%	12.30%	**0.0%**
CIFAR10	small	69.54%	69.50%	0.10%	**0.0%**
	deep	77.28%	75.13%	46.70%	**0.0%**

Our certified design workflow

- Our verification technique is an extension of VeryDiff
- It propagates the difference $|f(x) - \pi(x)|$ through the network
- Side effect: we can compute certified accuracy bounds!

Certified Private Inference: Cryptographic Noise

	size	N	QP	q_i	Δ	d_{num}	Poly deg.	Depth	CKKS error	Runtime
MNIST	4×256	2^{17}	2^{2291}	30	40	2	119	49	10^{-3}	31
	4×256	2^{17}	2^{2746*}	45	60	2	119	49	10^{-7}	39
	6×256	2^{17}	2^{3341}	30	40	2	119	72	10^{-3}	60
	6×256	2^{17}	2^{3841*}	45	60	7	119	72	10^{-7}	74
HELOC	$[64, 32]$	2^{16}	2^{1031}	30	40	2	27	21	10^{-5}	1
	$[64, 32]$	2^{16}	2^{1306}	45	60	2	27	21	10^{-9}	1
CIFAR-10	small	2^{17}	2^{2921}	30	40	2	119	62	10^{1}	78
	small	2^{17}	2^{3511}	45	60	5	119	62	10^{-4}	95
	small	2^{17}	2^{3101}	30	40	2	247	66	10^{-1}	98
	small	2^{17}	2^{3511}	45	60	7	247	66	10^{-4}	119
	deep	2^{17}	2^{2921}	30	40	2	119	62	10^{1}	135
	deep	2^{17}	2^{3511}	45	60	5	119	62	10^{-4}	167
	deep	2^{17}	2^{3101}	30	40	2	247	66	10^{-1}	155
	deep	2^{17}	2^{3511}	45	60	7	247	66	10^{-4}	191

Side note: encrypted inference introduces noise

- It is small, stochastic, but might still cause overflows
- Future work: probabilistic verification!

Summary

Neural network verification exists!

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

Applications to safe control

- Safety & liveness specs
- Quantisation and sampling

Applications to private inference

- FHE can only compute polynomials
- Differential verification for safe design

Any questions?