AutoML for Neural Network Robustness Verification

Matthias König | ADA Workshop on AutoAI

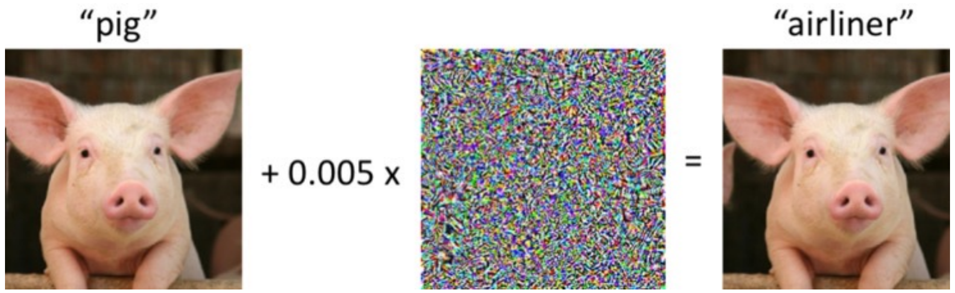
09.12.2022

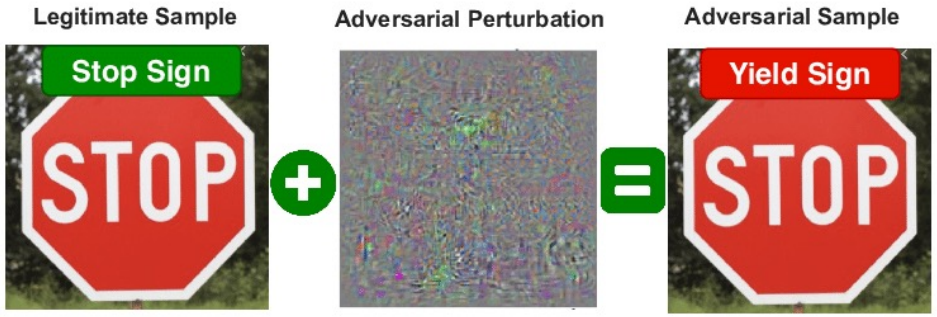
Universiteit Leiden
The Netherlands

Neural networks are vulnerable to adversarial examples

Neural networks are vulnerable to adversarial examples

Some examples… and possible consequences…





Verifying a deep neural network

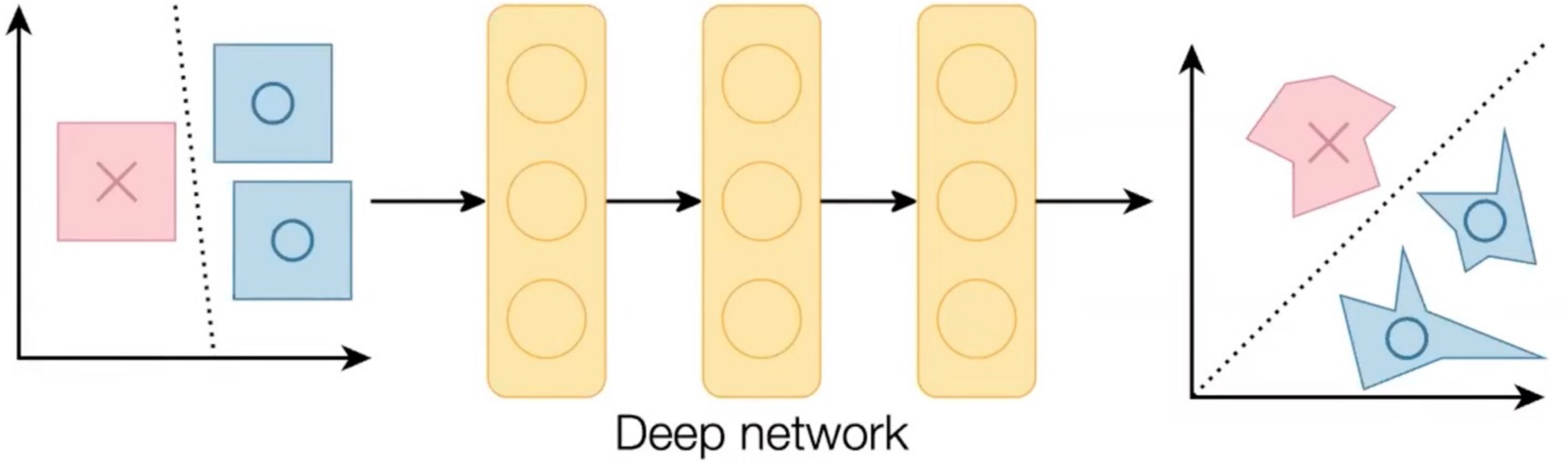
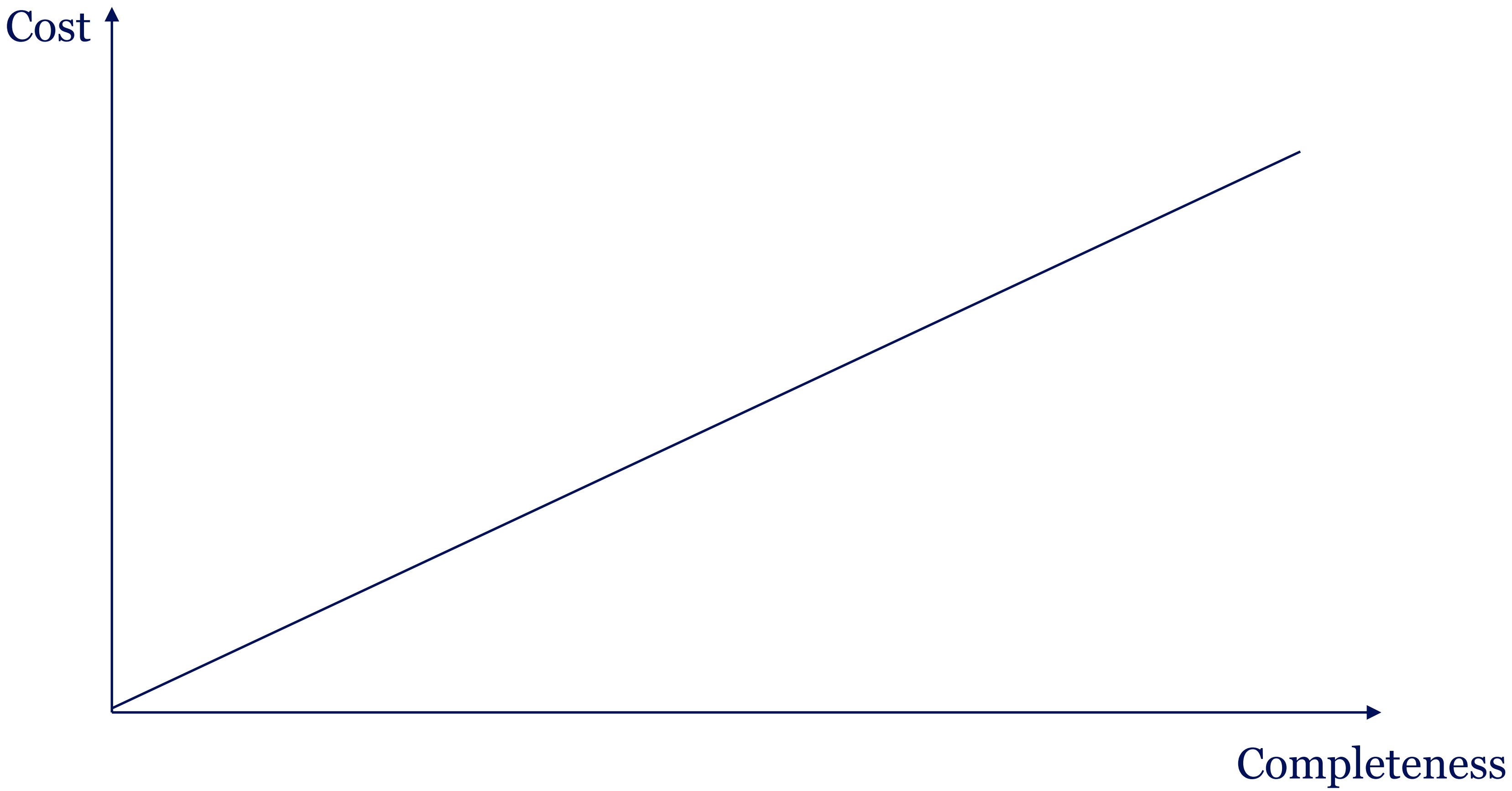


Image source: Stanford AI Safety Seminar

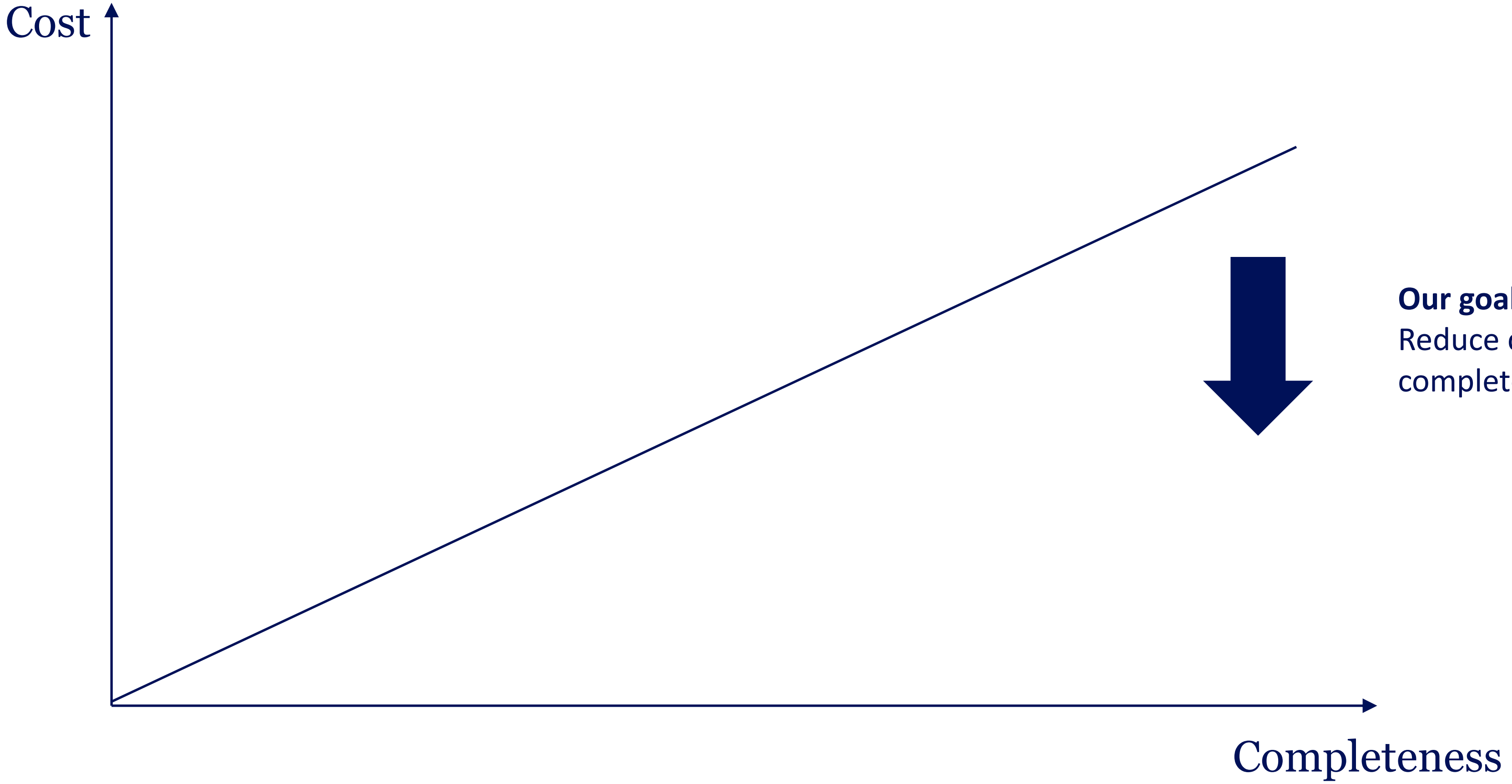
Neural network verification can be expensive

Incomplete vs. complete verification



Neural network verification can be expensive

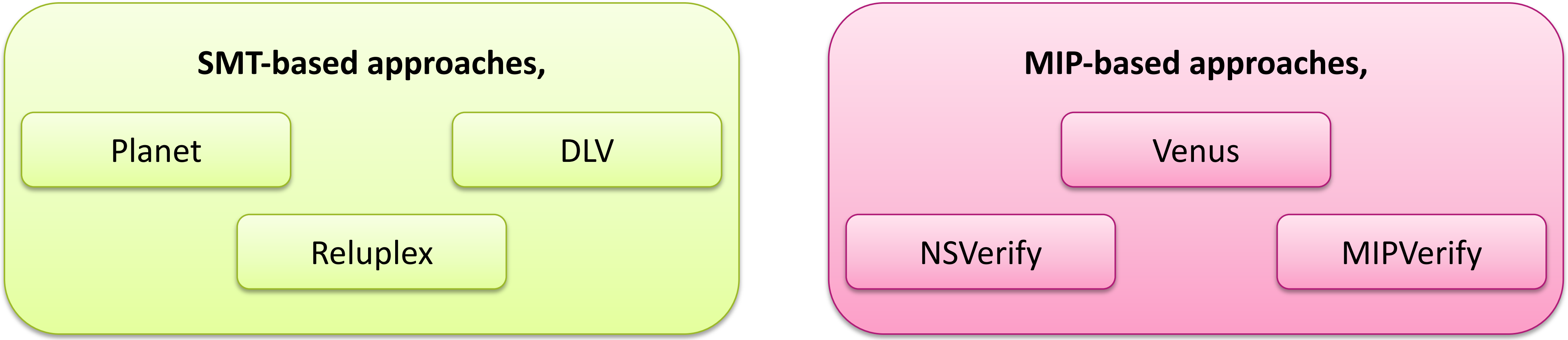
Incomplete vs. complete verification



Our goal:
Reduce computational cost of complete verification methods

There exist several approaches to verify a network

Some examples…



SMT-based approaches,

Planet

DLV

Reluplex

MIP-based approaches,

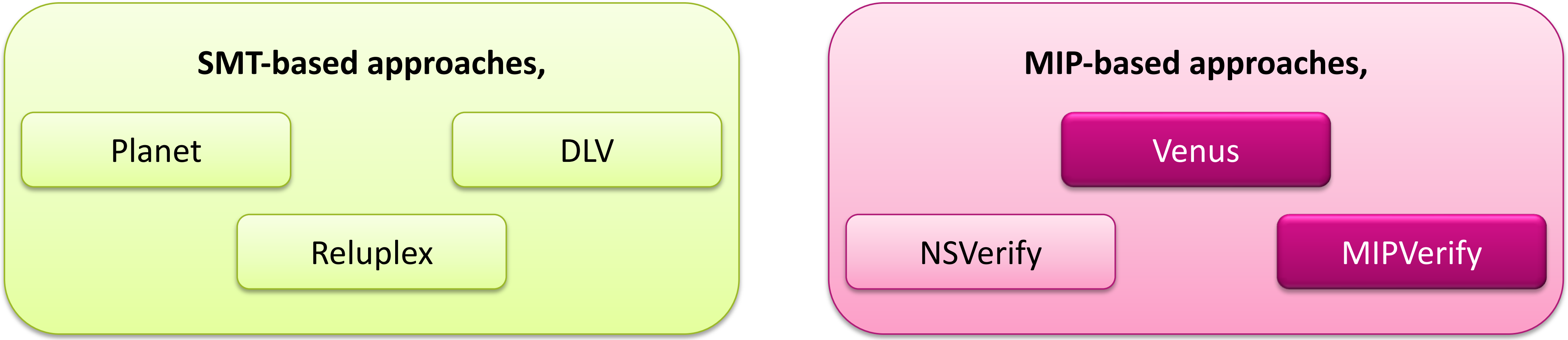
Venus

NSVerify

MIPVerify

There exist several approaches to verify a network

Some examples…



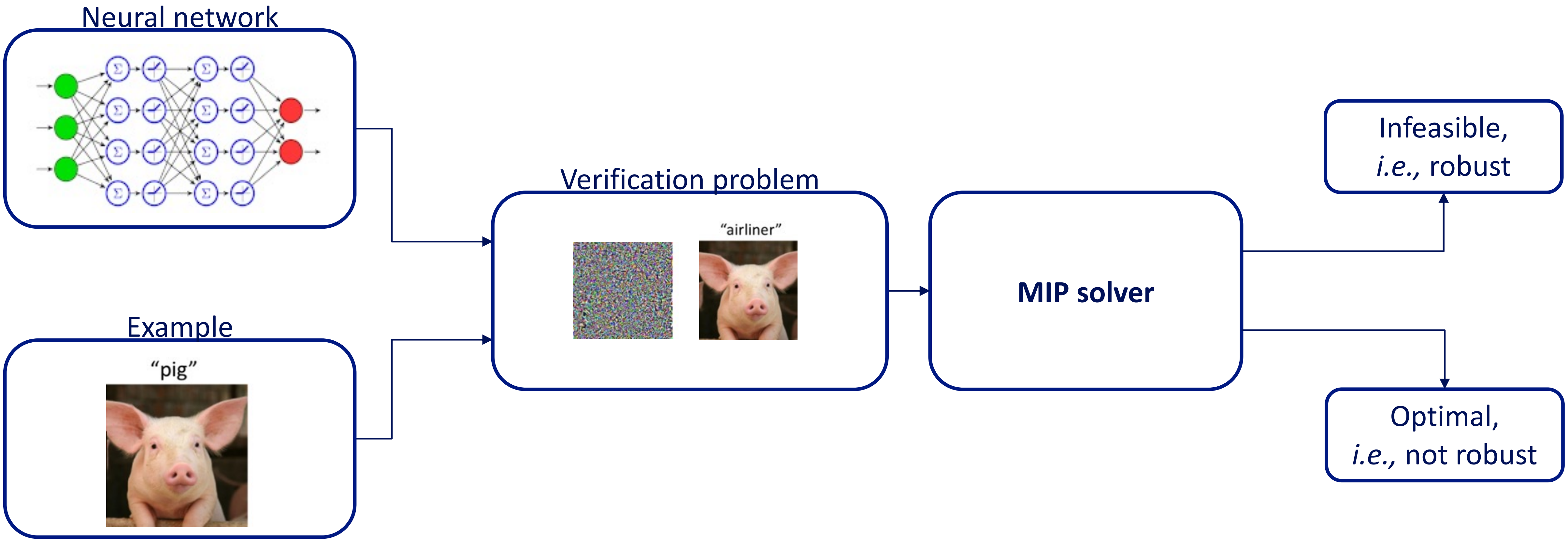
SMT-based approaches,

- Planet
- DLV
- Reluplex

MIP-based approaches,

- Venus
- NSVerify
- MIPVerify

General workflow of a MIP-based verifier



Neural network

Example

“pig”

Verification problem

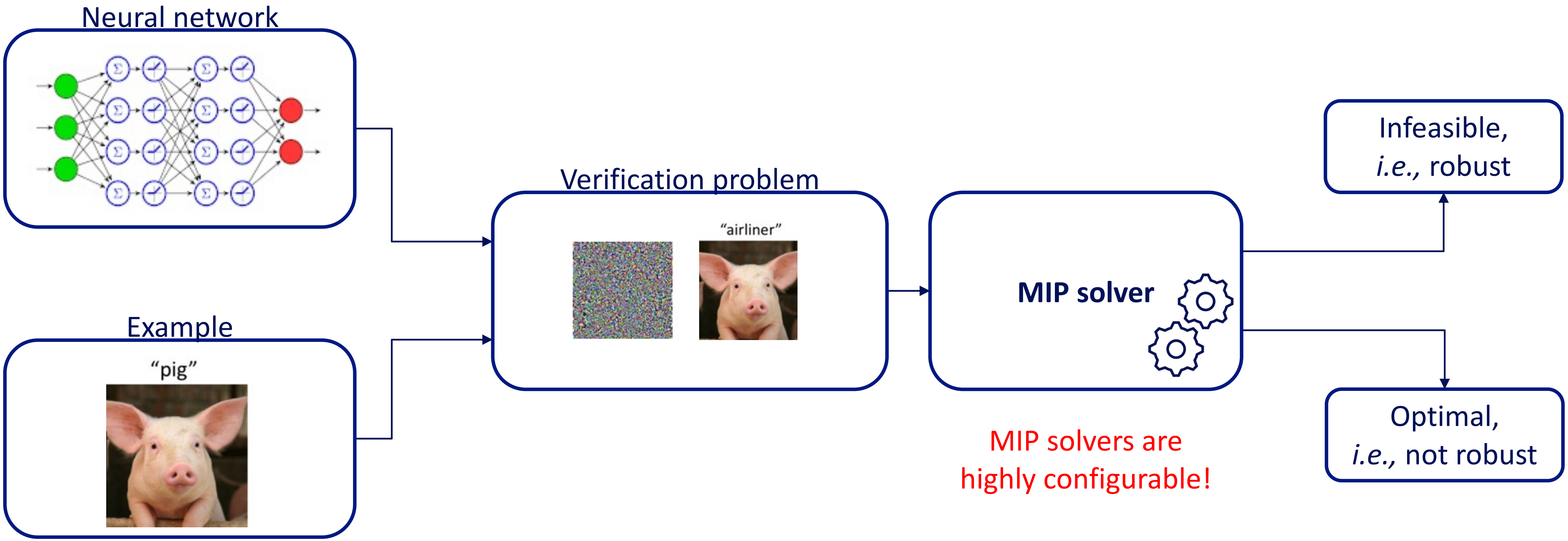
“airliner”

MIP solver

Infeasible,
i.e., robust

Optimal,
i.e., not robust

General workflow of a MIP-based verifier



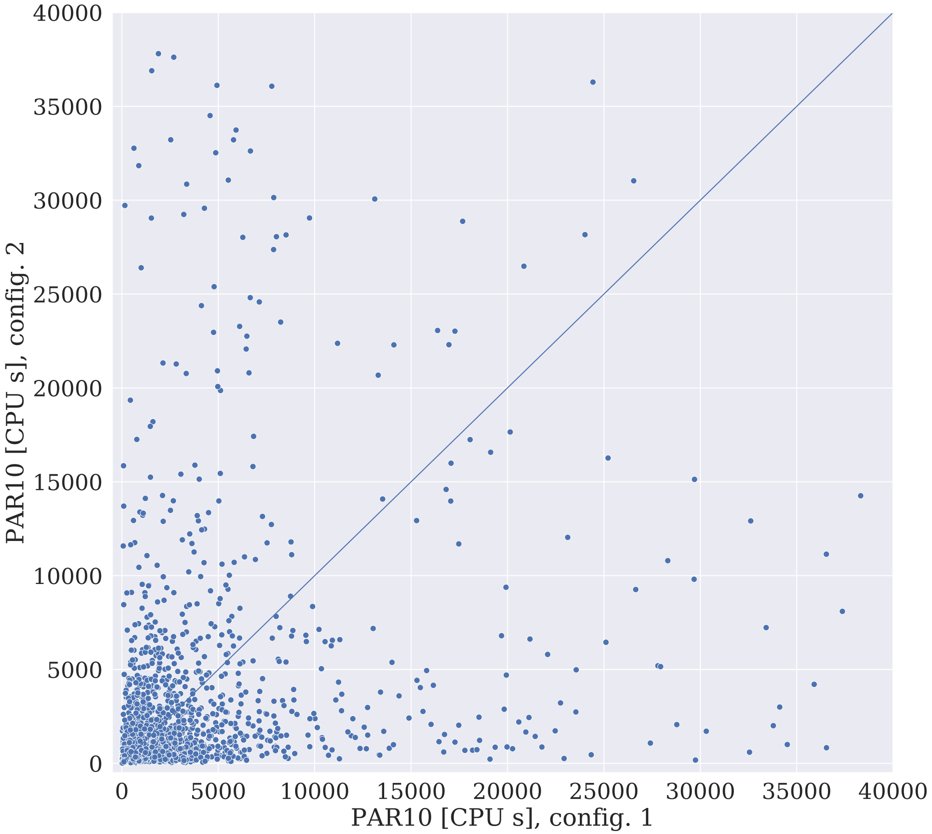
MIP solvers are highly configurable!

Main idea: Automated configuration of MIP solvers

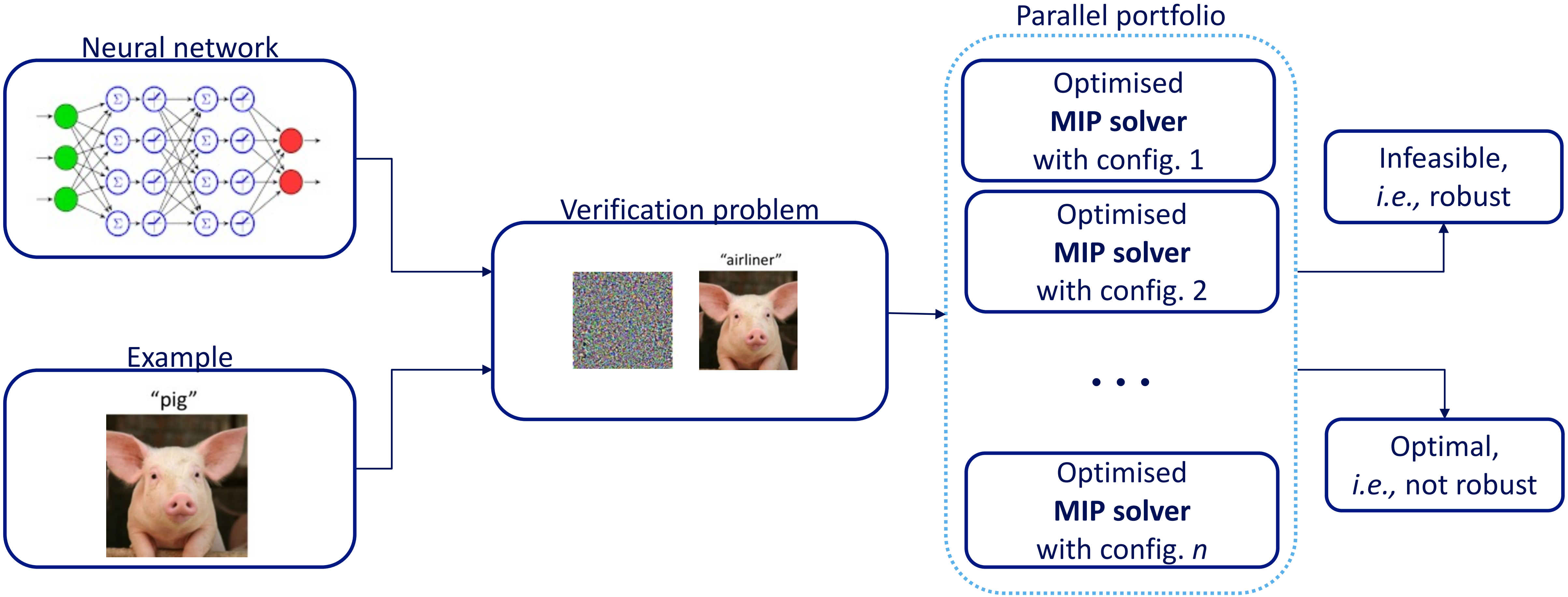
- Only succeeds if instance set is homogenous

Main idea: Automated configuration of MIP solvers

- Only succeeds if instance set is homogenous



Workflow of our proposed solution



Neural network

Example

“pig”

Verification problem

“airliner”

Parallel portfolio

Optimised **MIP solver** with config. 1

Optimised **MIP solver** with config. 2

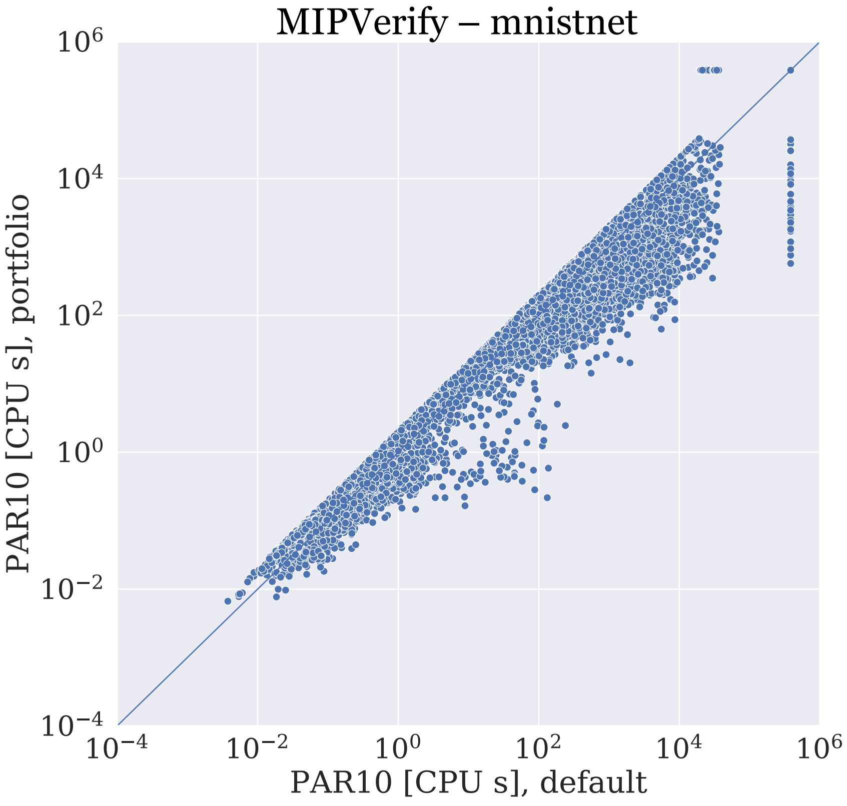
• • •

Optimised **MIP solver** with config. *n*

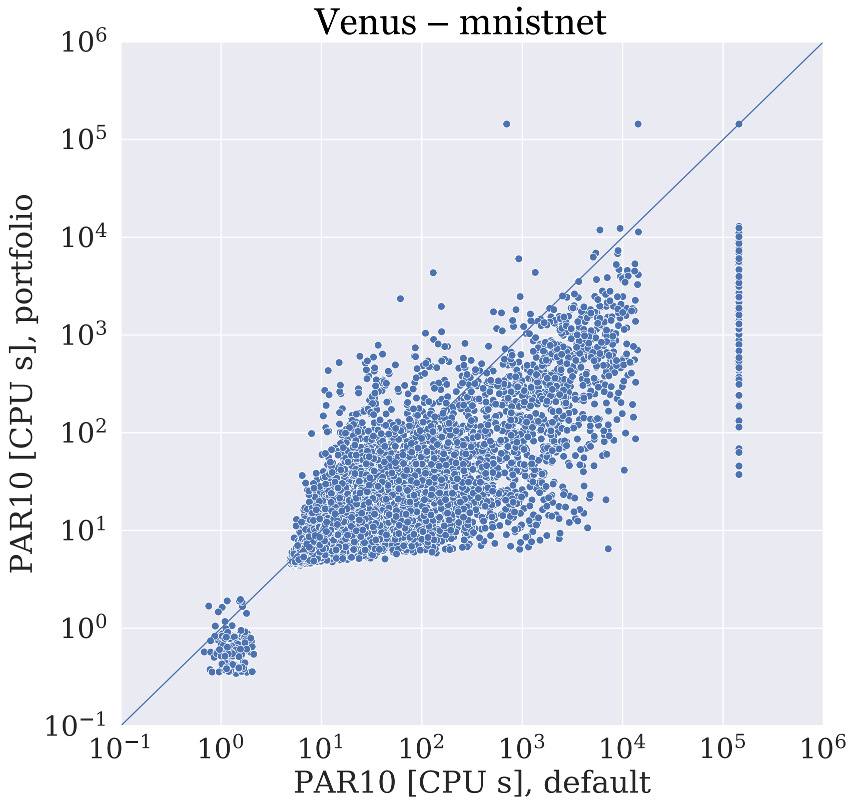
Infeasible, *i.e.,* robust

Optimal, *i.e.,* not robust

Our approach outperforms state-of-the-art approaches

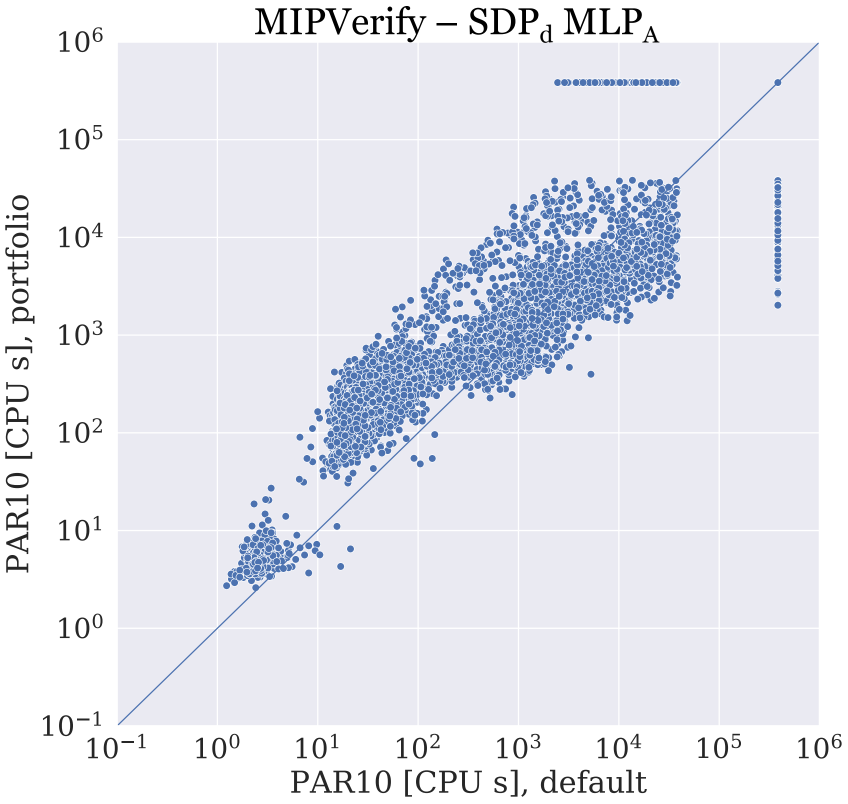


→ 1.61–fold speedup

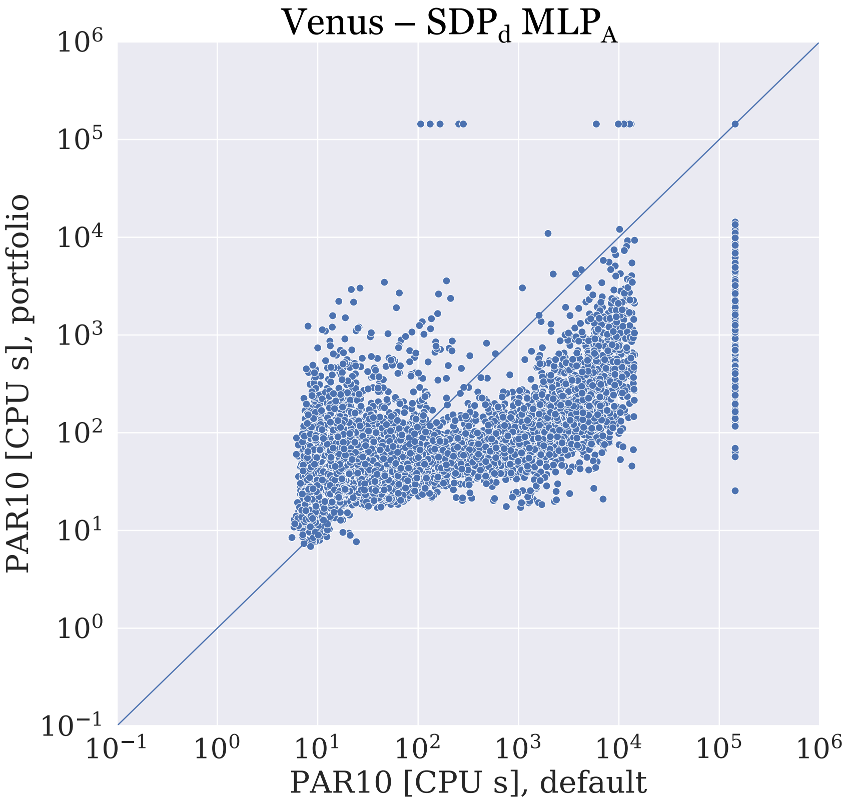


→ 7.26–fold speedup

Our approach outperforms state-of-the-art approaches



→ 4.7–fold speedup



→ 10.3–fold speedup

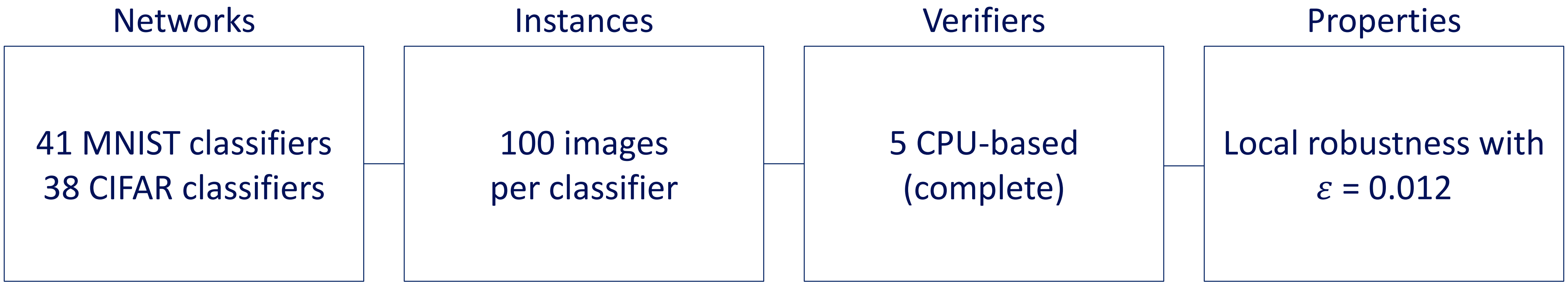
Conclusions

- **Automated algorithm configuration and portfolio construction** techniques can strongly improve the performance of neural network verification algorithms
- More specifically, we achieved substantial improvements over SOTA methods employed at default, in terms of CPU running time, timeouts and adversarial error bounds
- Future work involves automated selection and extension to further hyperparameters

[König, Hoos, van Rijn. Speeding Up Neural Network Robustness Verification via Algorithm Configuration and an Optimised Mixed Integer Linear Programming Solver Portfolio. *Machine Learning.* 2022.]

Does the observed heterogeneity of MIP-encoded verification problem instances generalise to other types of verification problem instances?

Critical assessment of neural network verifiers



Networks

41 MNIST classifiers
38 CIFAR classifiers

Instances

100 images
per classifier

Verifiers

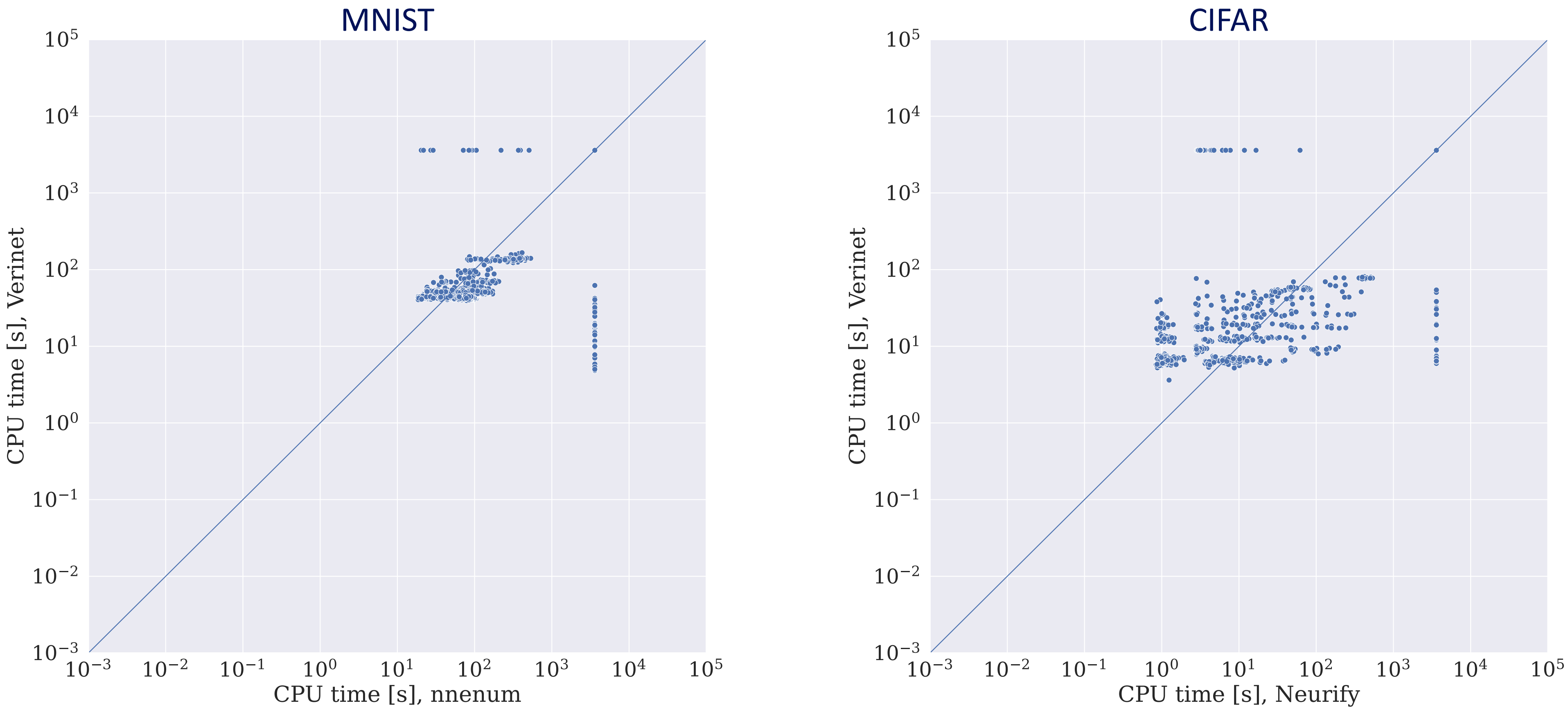
5 CPU-based
(complete)

Properties

Local robustness with
ε = 0.012

[König, Bosman, Hoos, van Rijn. Critically Assessing the State of the Art in CPU-based Local Robustness Verification. *Workshop on Artificial Intelligence Safety @AAAI*. 2023]

Critical assessment of neural network verifiers

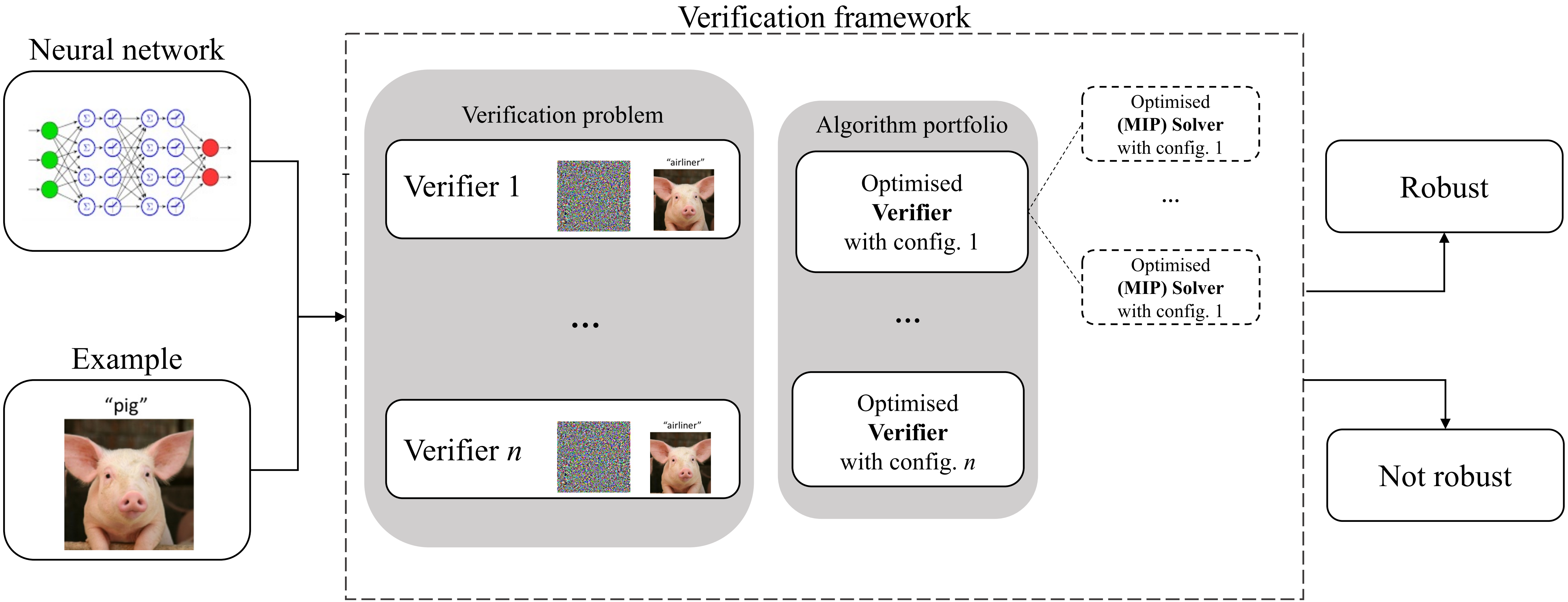


MNIST

CIFAR

[König, Bosman, Hoos, van Rijn. Critically Assessing the State of the Art in CPU-based Local Robustness Verification. *Workshop on Artificial Intelligence Safety @AAAI*. 2023]

Vision: Auto-Verify for neural network verification


Neural network
Example
"pig"
Verification framework
Verification problem
Verifier 1
"airliner"
...
Verifier n
"airliner"
Algorithm portfolio
Optimised **Verifier** with config. 1
...
Optimised **Verifier** with config. n
Optimised **(MIP) Solver** with config. 1
...
Optimised **(MIP) Solver** with config. 1
Robust
Not robust