DTU Wind

LORER Summer School 2025
Montreal, Canada

Machine Learning for Power Systems: Is it time to trust it?

Spyros Chatzivasileiadis
Professor
DTU Wind

Today's plan

1. Why Machine Learning?
2. The importance of high-quality data
3. Power systems are physical systems → Physics-Informed Neural Networks
4. Verify AI: worst-case guarantees for Neural Networks

Bonus! (according to time left and what you like us to talk about!)

i. Play with a PINN!
   - Google Colab Notebook to experiment with

ii. Integrate worst-case violations in NN training

iii. Graph Neural Networks

DTU Department of Wind and Energy Systems

Working for a sustainable future

Management

Head of Department
Morten Jeppesen

Administration

Head of Administration
Søren Knudsen

Wind Energy Materials & Components (WMC)

Head of Division
Ignacio Martí

Wind Turbine Design (WTD)

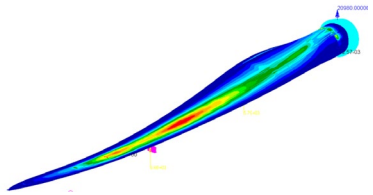
Head of Division
Kenneth Thomsen

Wind Energy Systems (WES)

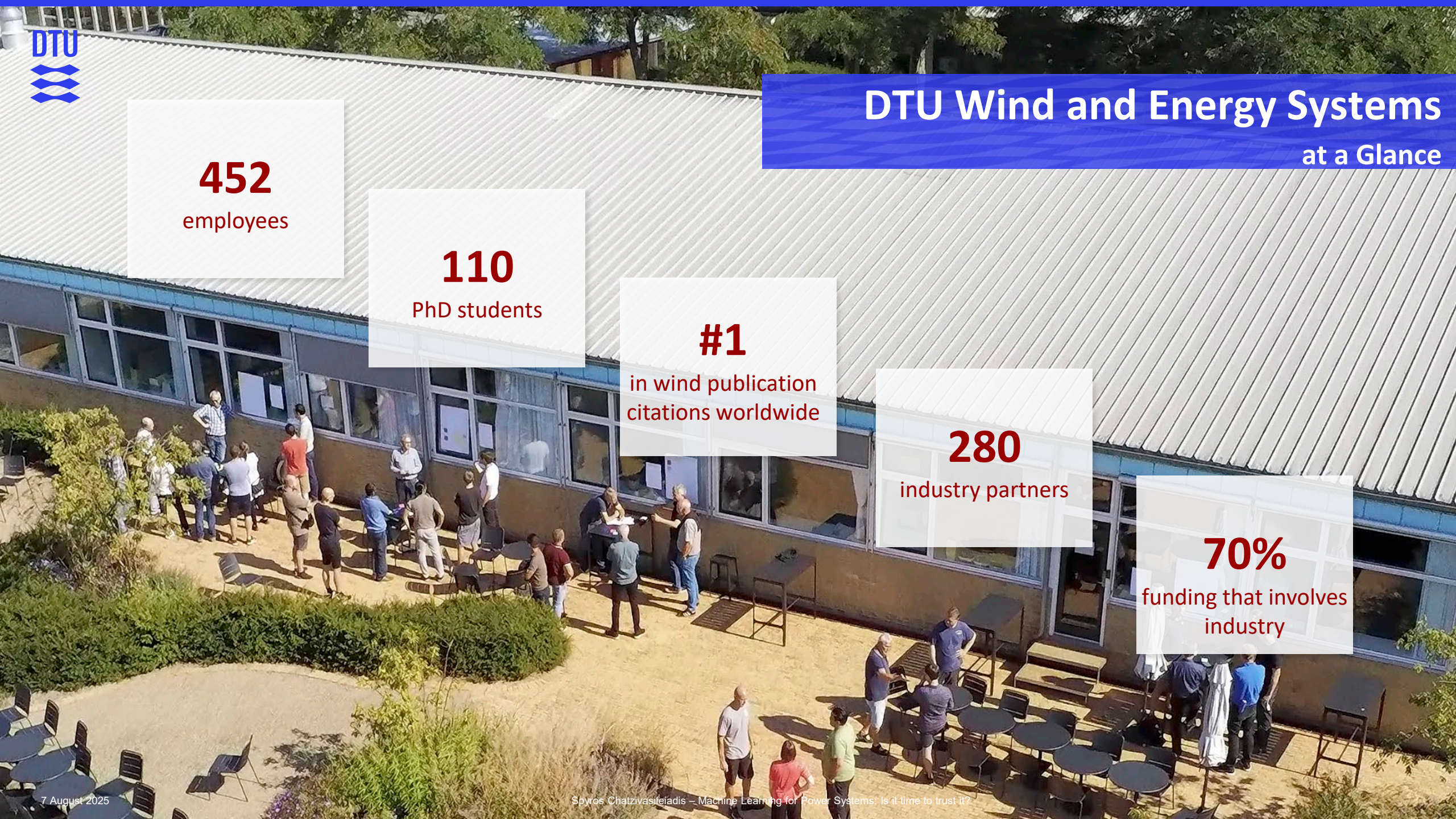
Head of Division
Lena Kitzing

Power and Energy Systems (PES)

Head of Division
Jacob Østergaard



DTU Wind and Energy Systems
at a Glance



452
employees

110
PhD students

#1
in wind publication citations worldwide

280
industry partners

70%
funding that involves industry

This work would not have been possible without the hard work of several people! Many thanks to…


Andreas Venzke

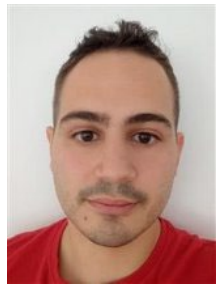

Rahul Nellikkath


Sam Chevalier


Lejla Halilbasic

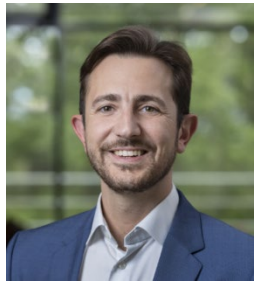

Elea Prat


Ilgiz Murzakhanov


Petros Ellinas


Agnes Nakiganda


Bastien Giraud


Spyros Chatzivasileiadis


Georgios Misyris


Florian Thams


Jochen Stiasny


Brynjar Sævarsson


Emilie Jong


Ignasi Ventura Nadal


Indrajit Chaudhuri

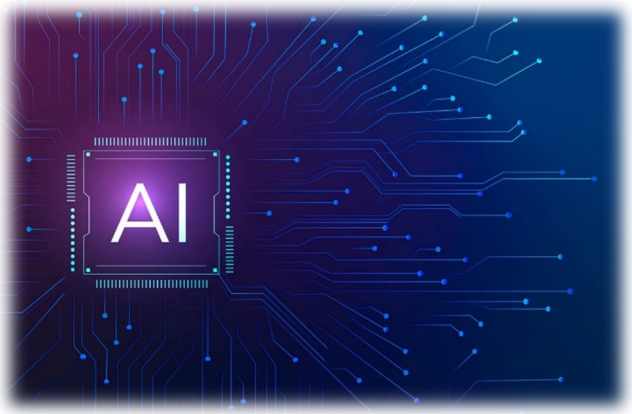
And to our collaborators:

Pascal van Hentenryck, Georgia Tech
Mathieu Tanneau, Georgia Tech
Dan Molzahn, Georgia Tech

Steven Low, Caltech
Guannan Qu, Caltech (now at CMU)

Baosen Zhang (Univ. Washington)

And many thanks to the European Research Council for funding this research



AI and Energy:
two of the Sectors with the highest growth potential

Machine learning: Why shall we apply it in power systems?

Machine learning: Why shall we apply it in power systems?

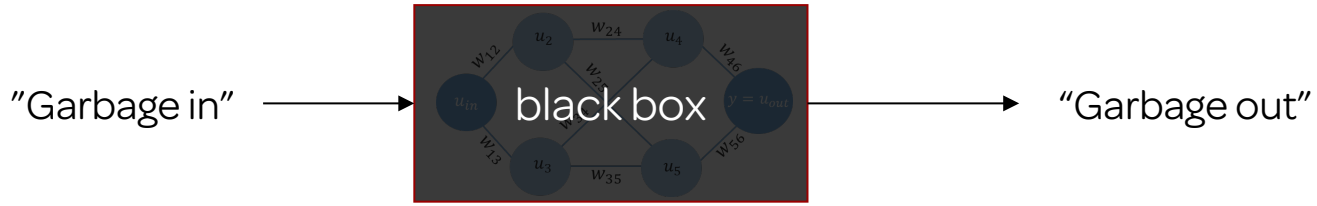
Machine Learning

1. ML methods can handle well extremely complex systems
2. ML methods can infer from incomplete data
3. ML methods can be extremely fast

Power Systems

1. Real-life power systems are described by thousands of variables, parameters, and differential-algebraic equations
2. It is computationally impossible (intractable) to check for all possible operating conditions
3. Build proxies (=surrogate models) → get an estimate 100-1'000 times faster than conventional models; assess 100-1'000 more scenarios in the same time

ML Barriers for Power systems



BUT:

1. Why would we use a "**black box**" to decide about **a safety-critical application**?

2. **Accuracy is** a **purely statistical** performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?

3. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

Takeaway #1

Solid motivation is key: If you wish to apply machine learning (including deep learning) methods on any problem, develop **solid arguments** why this is the only or the best way to do it

Machine learning: WHEN shall we apply it in power systems?
(i.e. what are the use cases?)

Machine learning: When shall we apply it in power systems? (my view)

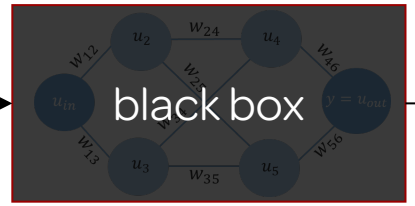
1. **When there is no other option, e.g. forecasting**
   - Load Forecasting has been the first real application for AI in power systems. Now also used for wind and olar forecasting, price forecasting, predictive maintenance, and others.
   - Very complex: No physics-based model can capture the interdependency between all variables

2. **When computation speed is important:** ML can be **100x-1000x faster**
   - Power system security: assess 1'000 critical scenarios in the same time conventional methods assess only 1
   - ("real-time") Energy markets: assess very fast possible options and determine best bidding strategy
   - **Added benefit: once trained, ML methods can run on a laptop** (no need to continuously using High-Performance Computing every time we want to assess some additional scenarios)

ML Proxies

Extremely fast, and hopefully accurate

ML Barriers for Power systems

"Garbage in" → 

→ "Garbage out"

High Quality Training Data

1. Why would we use a "**black box**" to decide about **a safety-critical application**?

2. **Accuracy is** a **purely statistical** performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?

3. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

⇨ **Neural Network Performance Guarantees**

→ *Remove dependence on the test database*

⇨ **Physics-Informed Neural Networks**

→ *Prior knowledge!*

Goals of this lecture

1. The importance of high-quality data
2. Sampling beyond statistics
3. Physics-Informed Neural Networks
4. Neural network verification

---

5. If time permits (appendix):
   - Graph Neural Networks for N-k Contingency Assessment

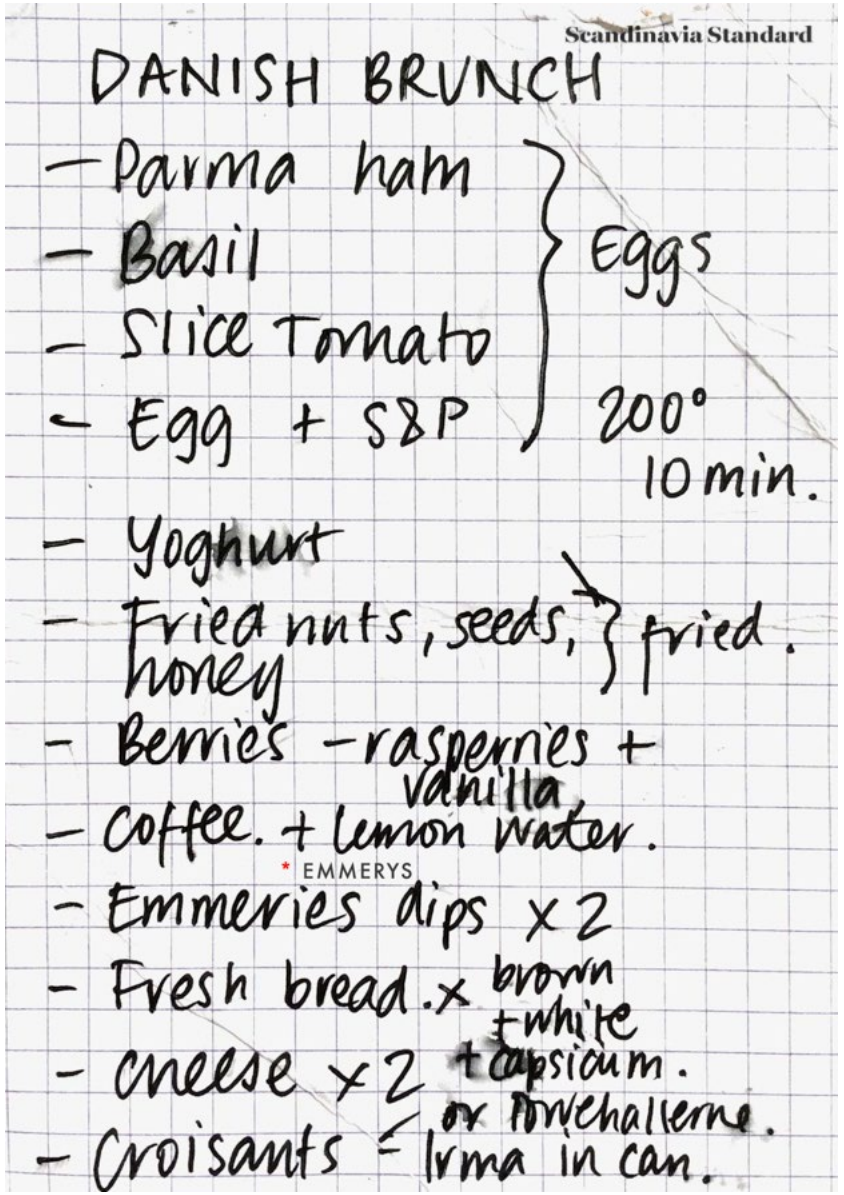
Article without any equations ☺

S. Chatzivasileiadis, A. Venzke, J. Stiasny and G. Misyris, **"Machine Learning in Power Systems: Is It Time to Trust It?,"** in *IEEE Power and Energy Magazine*, vol. 20, no. 3, pp. 32-41, May-June 2022 [.pdf]

Machine learning applications
(for power system security assessment)
A very short overview

The ingredients

- Data #1: A training database
- A training algorithm
- Data #2: A test database
  - To test accuracy of the approach



Test Database

Test Database

Traditionally:

- Split training database to e.g. 80% training samples and 20% test samples
- Train with the 80%
- Test with the 20%

Modern toolboxes have this integrated and automatized →only need to provide a training database

Point to remember:

The test database determines the performance of your method. If the test data come from the same simulations as your training data, the accuracy can be deceivingly high. Would it be equally high in reality?

Ideally → use a different real-life dataset

(Unfortunately, not always possible)

Takeaway #2

The quality of your test database is crucial: the test database determines the performance of your method; for a valid assessment, it needs to include a wide range of operating conditions with the same frequency of occurrence as in real-life

Evaluating the performance of a neural network

Performance Metrics: Accuracy

	Actually Safe	Actually Unsafe
Predicted safe	True positive (TP)	False positive (FP)
Predicted Unsafe	False negative (FN)	True Negative (TN)

- Accuracy: The proportion of correct classifications in the whole dataset

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

- **Example:** Assume 1000 datapoints

Actually safe: 500	Actually unsafe: 500
TP=480	FP=30
FN=20	TN=470

Accuracy = ?

Evaluating performance by measuring only accuracy is often not enough

Why?

Performance Metrics: Accuracy

	Actually Safe	Actually Unsafe
Predicted safe	True positive (TP)	False positive (FP)
Predicted Unsafe	False negative (FN)	True Negative (TN)

- Accuracy: The proportion of correct classifications in the whole dataset

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

- **Example:** Assume 1000 datapoints

Actually safe: 500	Actually unsafe: 500
TP=480	FP=30
FN=20	TN=470

$$\text{Accuracy} = \frac{480+470}{480+20+470+30} = 95\%$$

Evaluating performance by measuring only accuracy is often not enough

Why?

Performance Metrics: Accuracy

	Actually Safe	Actually Unsafe
Predicted safe	True positive (TP)	False positive (FP)
Predicted Unsafe	False negative (FN)	True Negative (TN)

- Accuracy: The proportion of correct classifications in the whole dataset

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

- **Example:** Assume 1000 datapoints

Actually safe: 500	Actually unsafe: 500
TP=480	FP=30
FN=20	TN=470

$$\text{Accuracy} = \frac{480+470}{480+20+470+30} = 95\%$$

Evaluating performance by measuring only accuracy is often not enough

Why?

Actually safe: 20	Actually unsafe: 980
TP=1	FP=30
FN=19	TN=950

Accuracy = ?

Performance Metrics: Accuracy

	Actually Safe	Actually Unsafe
Predicted safe	True positive (TP)	False positive (FP)
Predicted Unsafe	False negative (FN)	True Negative (TN)

- Accuracy: The proportion of correct classifications in the whole dataset

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

- **Example:** Assume 1000 datapoints

Actually safe: 500	Actually unsafe: 500
TP=480	FP=30
FN=20	TN=470

$$\text{Accuracy} = \frac{480+470}{480+20+470+30} = 95\%$$

Evaluating performance by measuring only accuracy is often not enough

Why?

Actually safe: 20	Actually unsafe: 980
TP=1	FP=30
FN=19	TN=950

$$\text{Accuracy} = \frac{1+950}{1+19+950+30} = 95\%$$

- 95% accurate but we have misclassified almost all truly safe points!
- For heavily unbalanced data, accuracy is not sufficient!

Performance metrics:
If you train a classifier, make sure you not only assess its performance based on accuracy

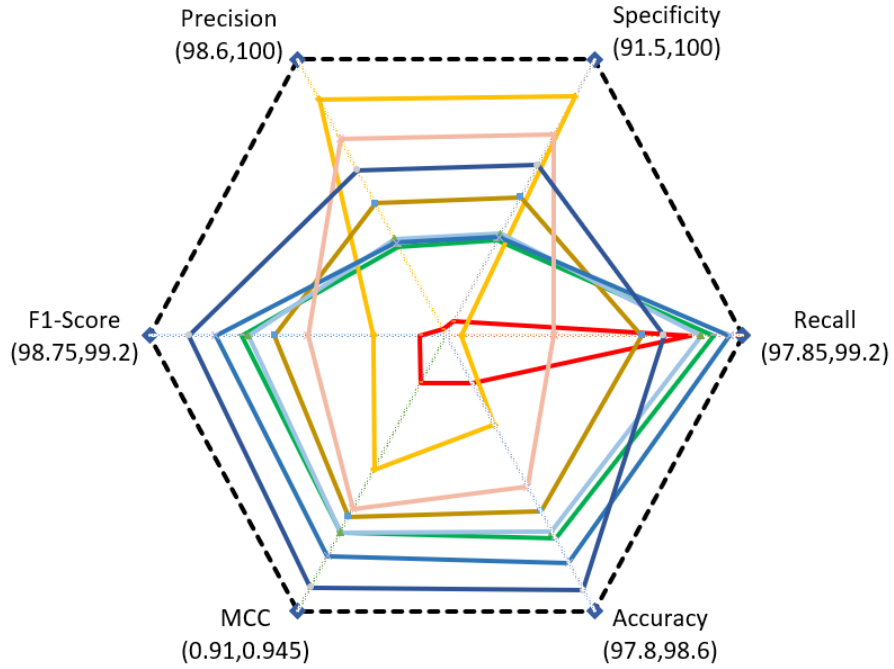
- Accuracy
- Recall: True Positive Rate $\frac{TP}{TP+FN}$
- Specificity: True Negative Rate $\frac{TN}{TN+FP}$
- Precision: Positive Predictive Value $\frac{TP}{TP+FP}$
- F1: harmonic mean of Precision and Recall $\mathrm{F1} = \frac{Precision \cdot Recall}{Precision+Recall}$
- MCC (Matthews correlation coefficient)

  (only for binary classification – 2 classes only)

  – MCC=1 → perfect prediction

  – MCC=0 → random (like flipping a coin)

  – MCC= –1→ Completely mistaken



Hidalgo-Arteaga, Hancharou, Thams, Chatzivasileiadis, Powertech 2019

$$\mathrm{MCC} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Key hints for your implementation

- Regularization: Training of neural networks work better if you normalize your inputs
  - Try to normalize your active power setpoints (e.g. if PG1 =30 MW and PG1max = 100 MW, then PG1=0.3)

- 1-hot encoding: Neural networks work better if you use *one vector for each class*

Instead of:

NN has 1 output neuron

Operating point	Safe=1 Unsafe=0
x_1	0
x_2	1
x_3	0

NN has 2 output neurons

Operating point	Safe	Unsafe
x_1	0	1
x_2	1	0
x_3	0	1

Takeaway #3:
Neural networks (or Decision Trees) for classification: you need a balanced training database → similar number of safe and unsafe points

Takeaway #4:
Accuracy is not sufficient to assess the NN/DT performance. We need additional metrics

Takeaway #5:
Neural Network training requires additional “tricks” to boost its performance (e.g. 1-hot encoding/regularization)

Training database
Sampling Beyond Statistics

B. Giraud , L. Charles, A. M. Nakiganda, J. Vorwerk, S. Chatzivasileiadis, A Dataset Generation Toolbox for Dynamic Security Assessment: On the Role of the Security Boundary, Sustainable Energy, Grids, and Networks, Elsevier, 2025, https://arxiv.org/pdf/2501.09513

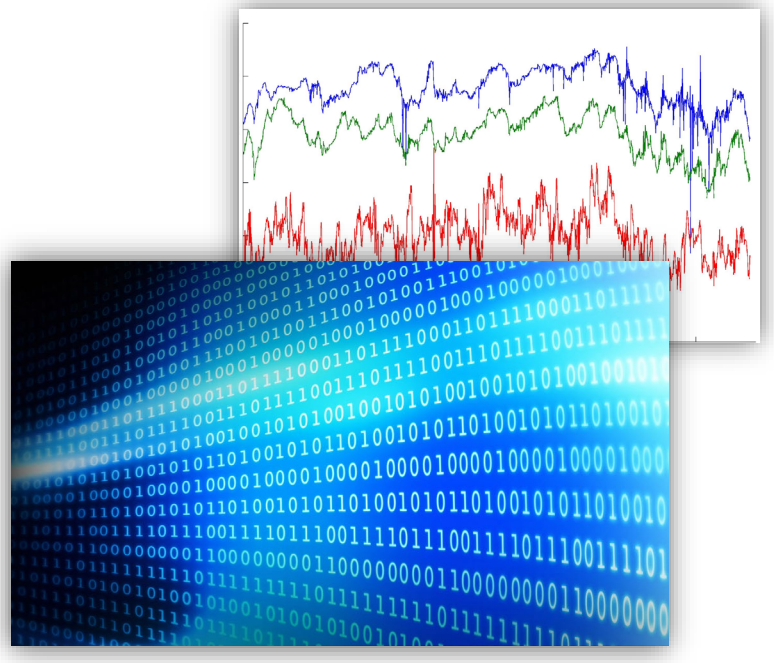
F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems". ". IEEE Trans. Power Systems, vol. 35, no. 1, pp. 30-41, Jan. 2020. https://www.arxiv.org/abs/1806.0107.pdf

We need data!

1. Historical data are often insufficient

2. We often need to generate our own data to test the performance of our ML algorithm before deployment (“emulate”)

3. **Need to generate simulation data**

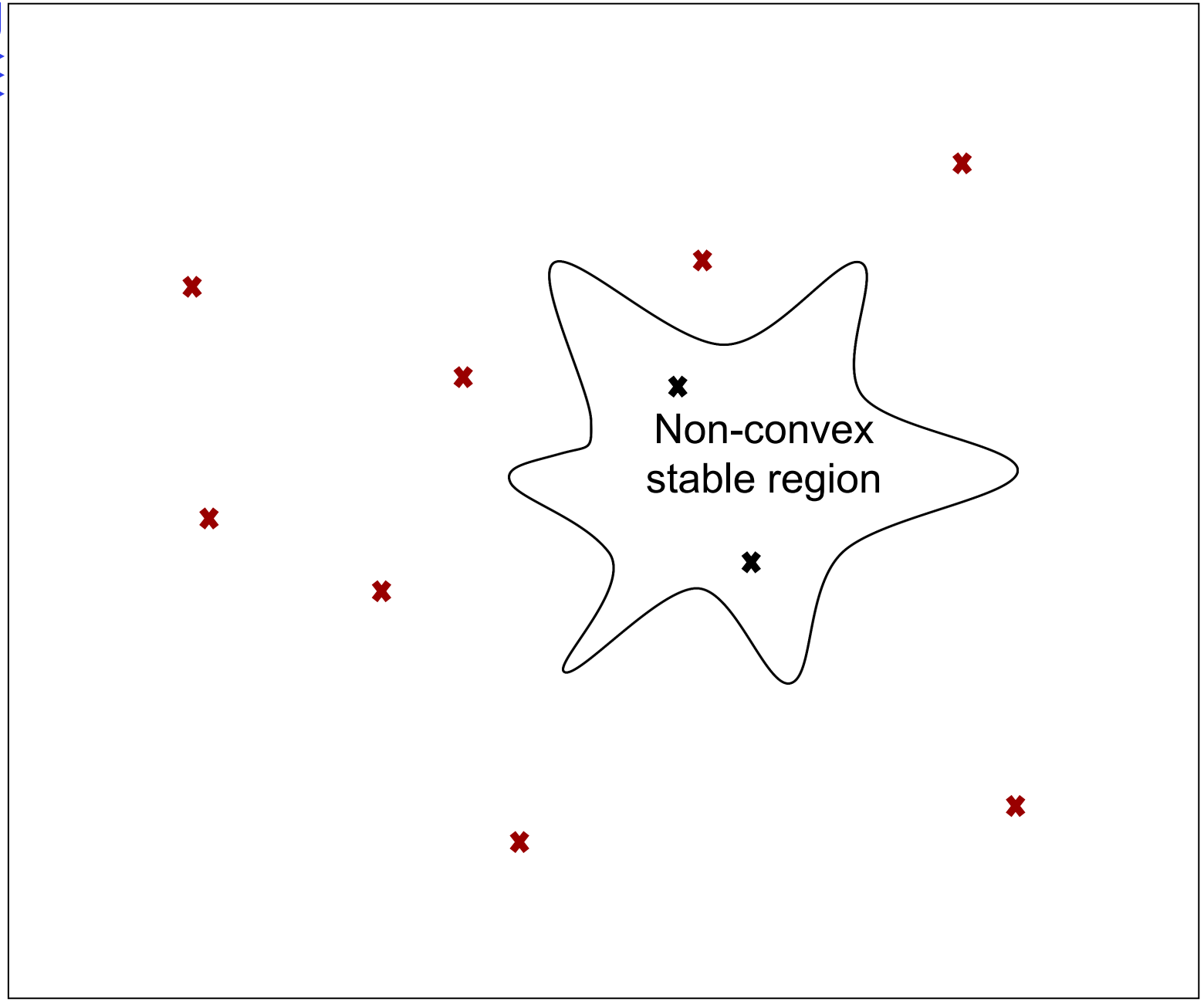


Here: generate data for power system security assessment

- Assessing the stability of 100’000s of operating points is an extremely demanding task
  - Immense search space
  - How can I do it efficiently?

Sampling beyond Statistics: Efficient Database Generation Toolbox

- Modular and highly efficient algorithm
- Can accommodate numerous definitions of power system security (e.g. N-1, N-k, small-signal stability, voltage stability, transient stability, **or a combination** of them)
- **10-50 times faster** than existing state-of-the-art approaches
- Our use case: N-1 security + small-signal stability



Conventional Database Generation

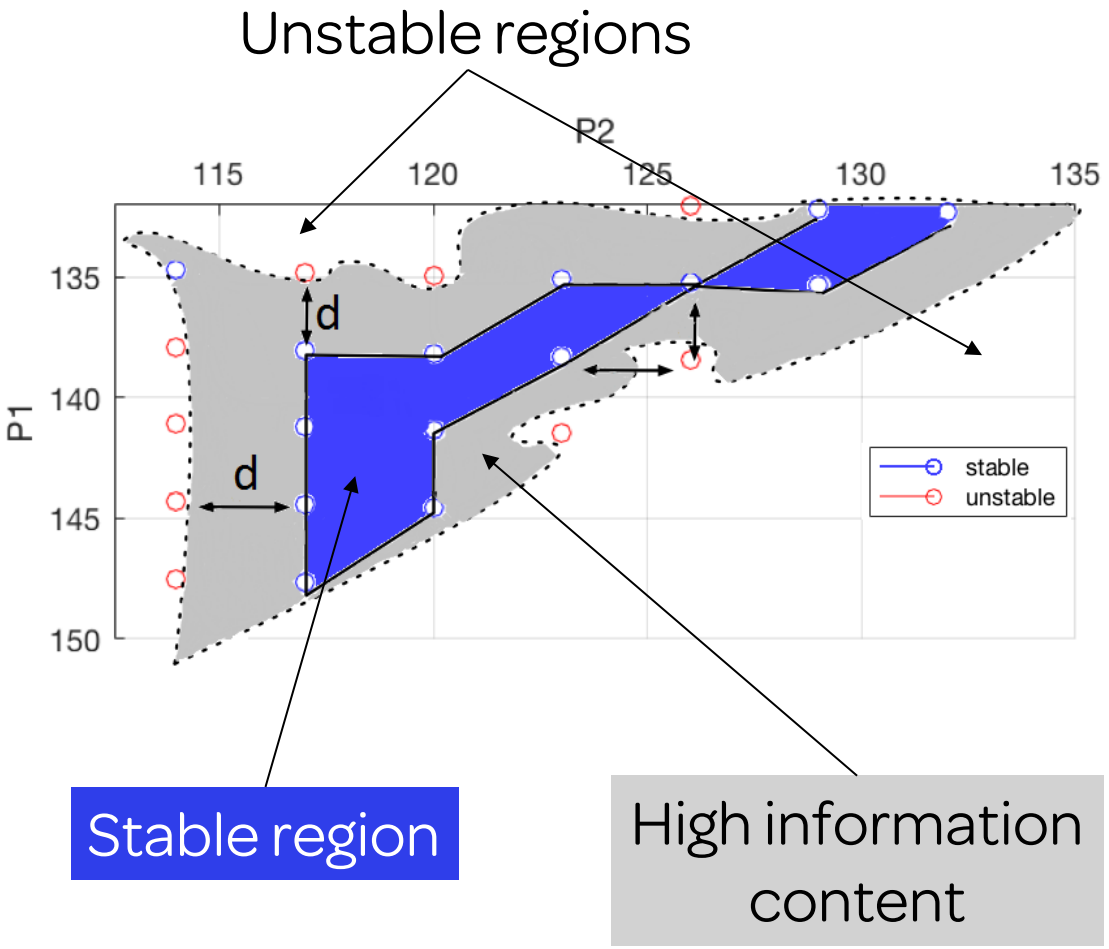
1. Statistical sampling across the input space

2. Often results to highly unbalanced database
   - the stable/safe region is often1%-2% of the total region

Alternative: use our prior knowledge

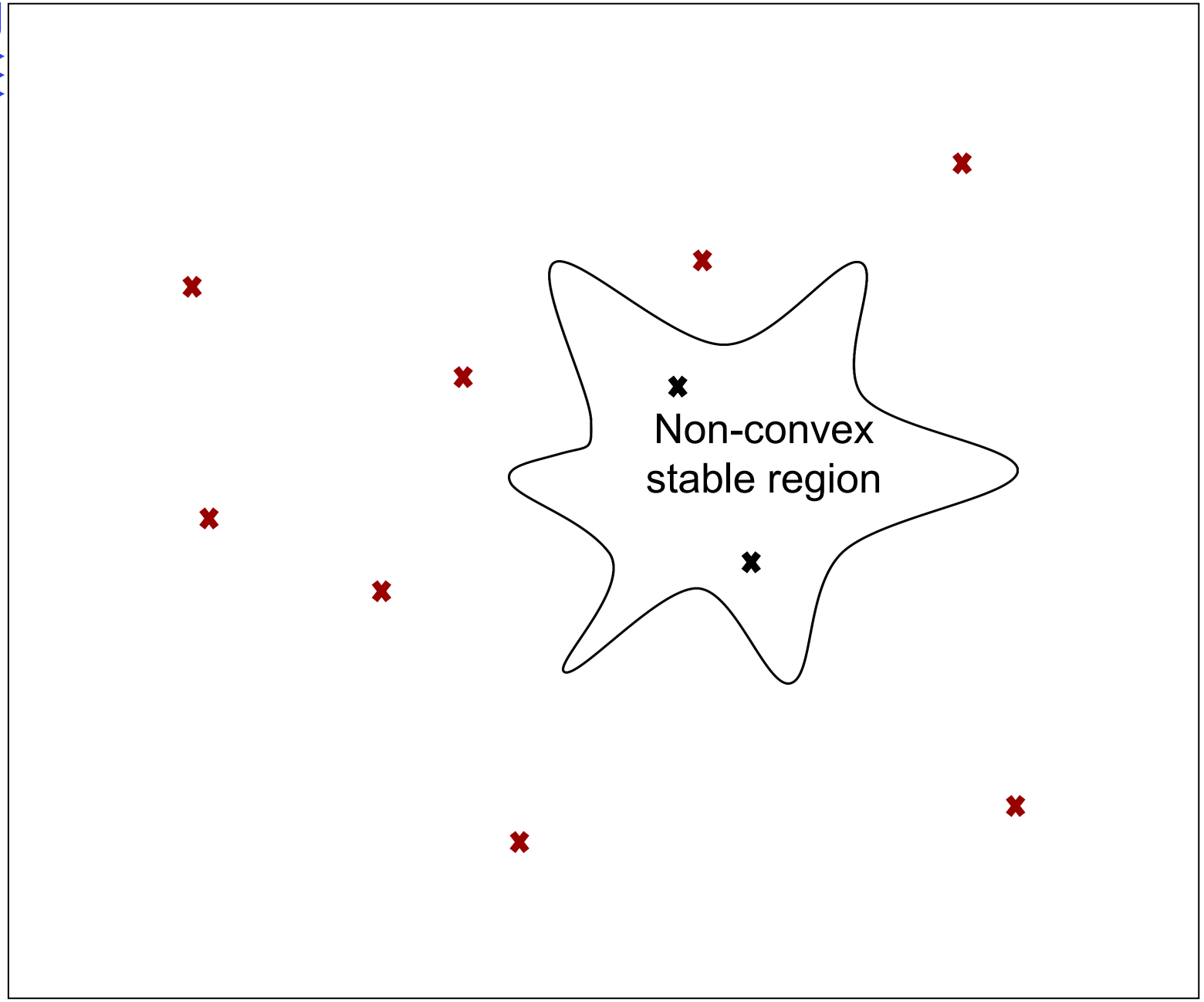
Sampling beyond Statistics: Efficient Database Generation

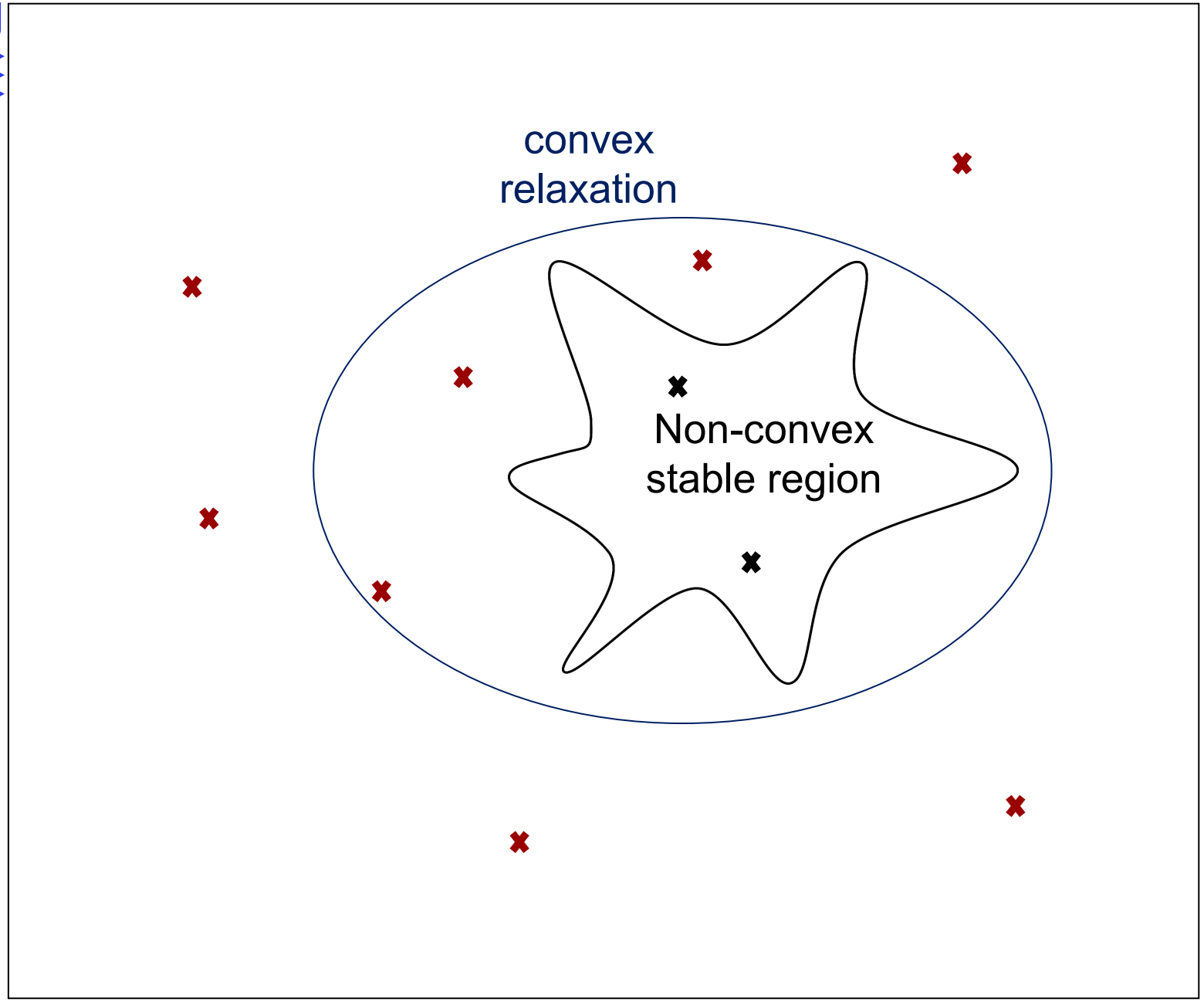
- The goal
  - Focus on the boundary between stability and instability
  - We call it: "high information content" region
- How?
  1. Using convex relaxations
  2. And "Directed Walks"



Real data for the IEEE 14-bus system
N-1 security and small-signal stability

Convex relaxations to discard infeasible regions

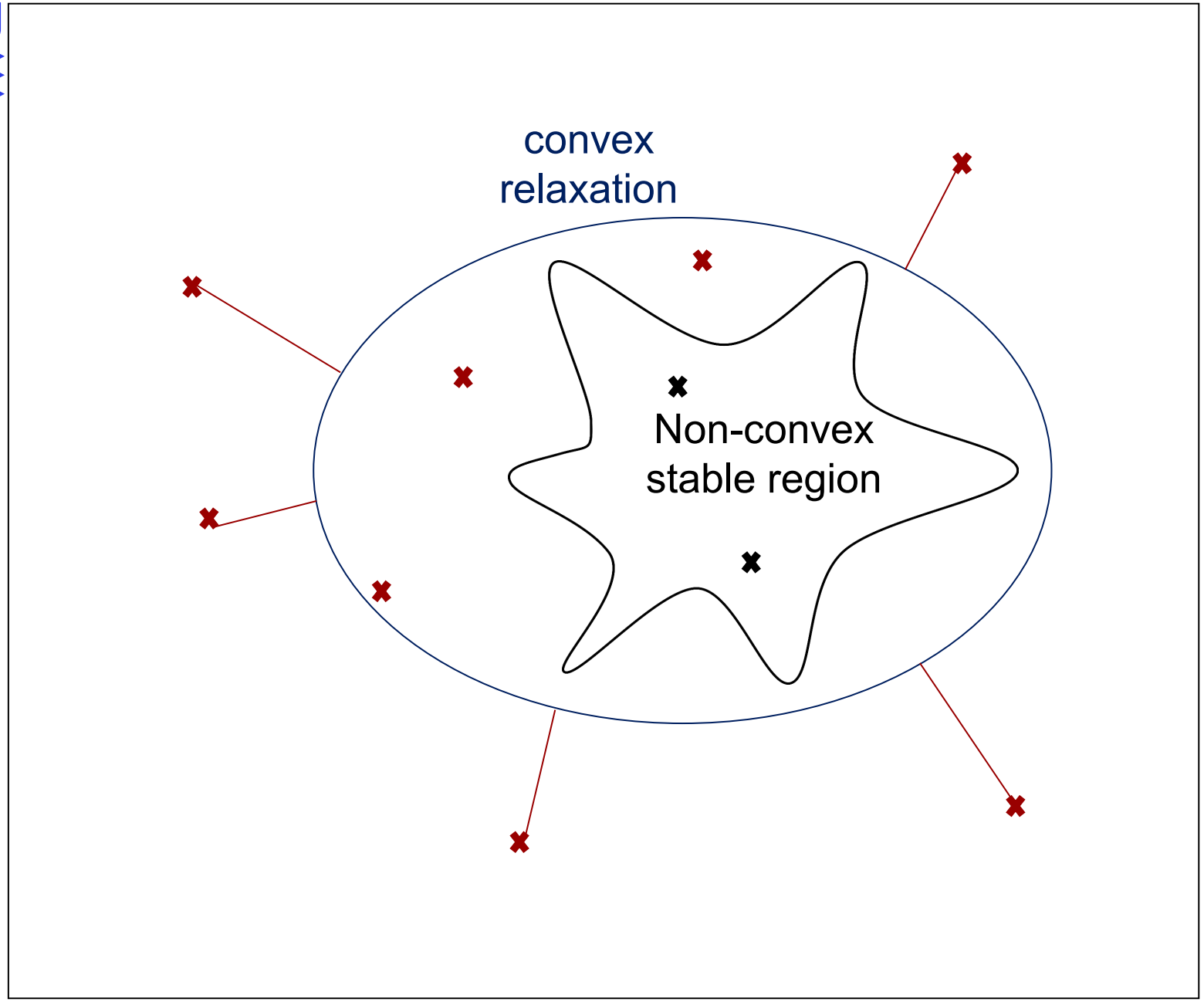




Convex relaxations to discard infeasible regions

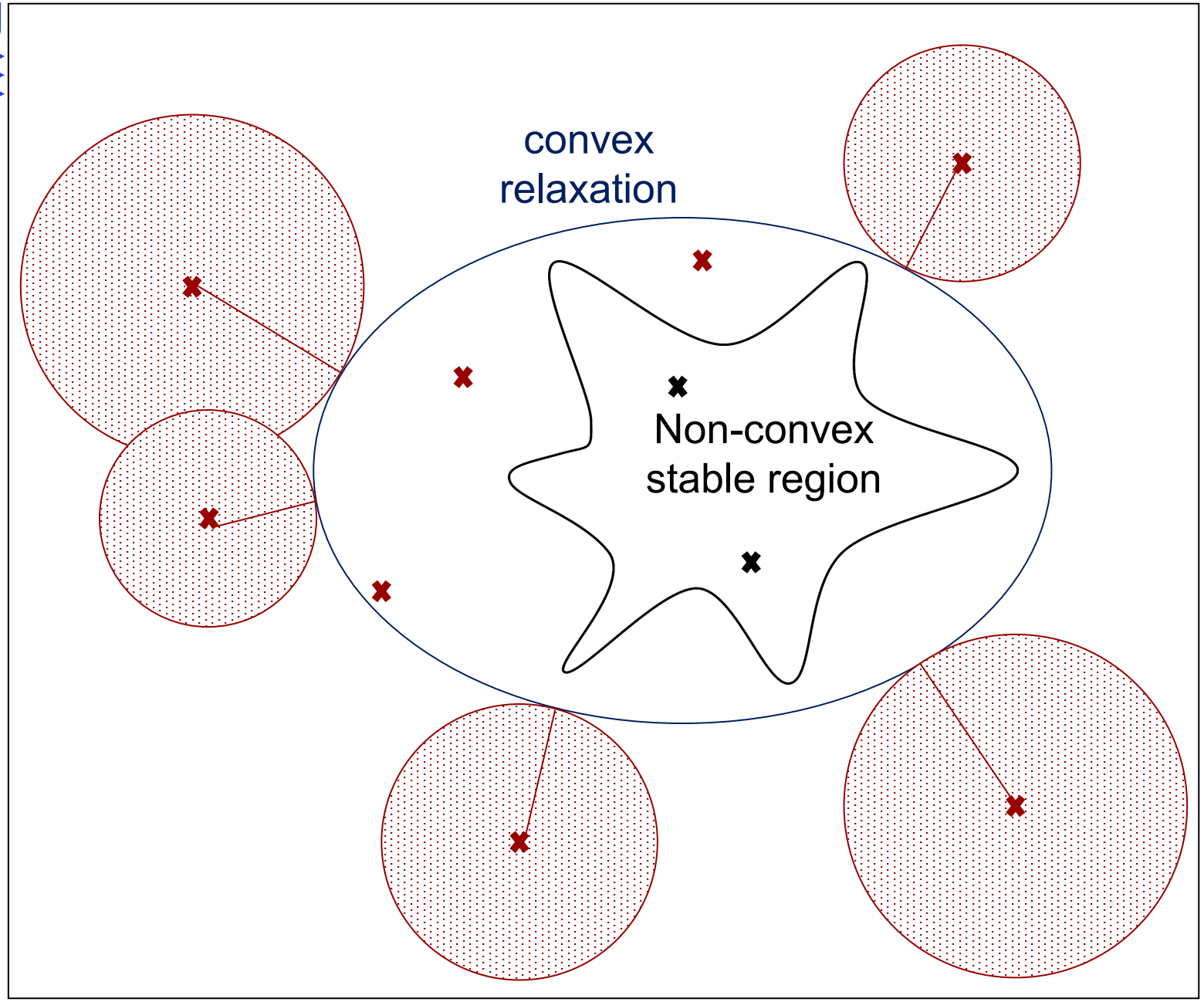
- Certificate: if point infeasible for semidefinite relaxation → infeasible for the original problem

Convex relaxations to discard infeasible regions

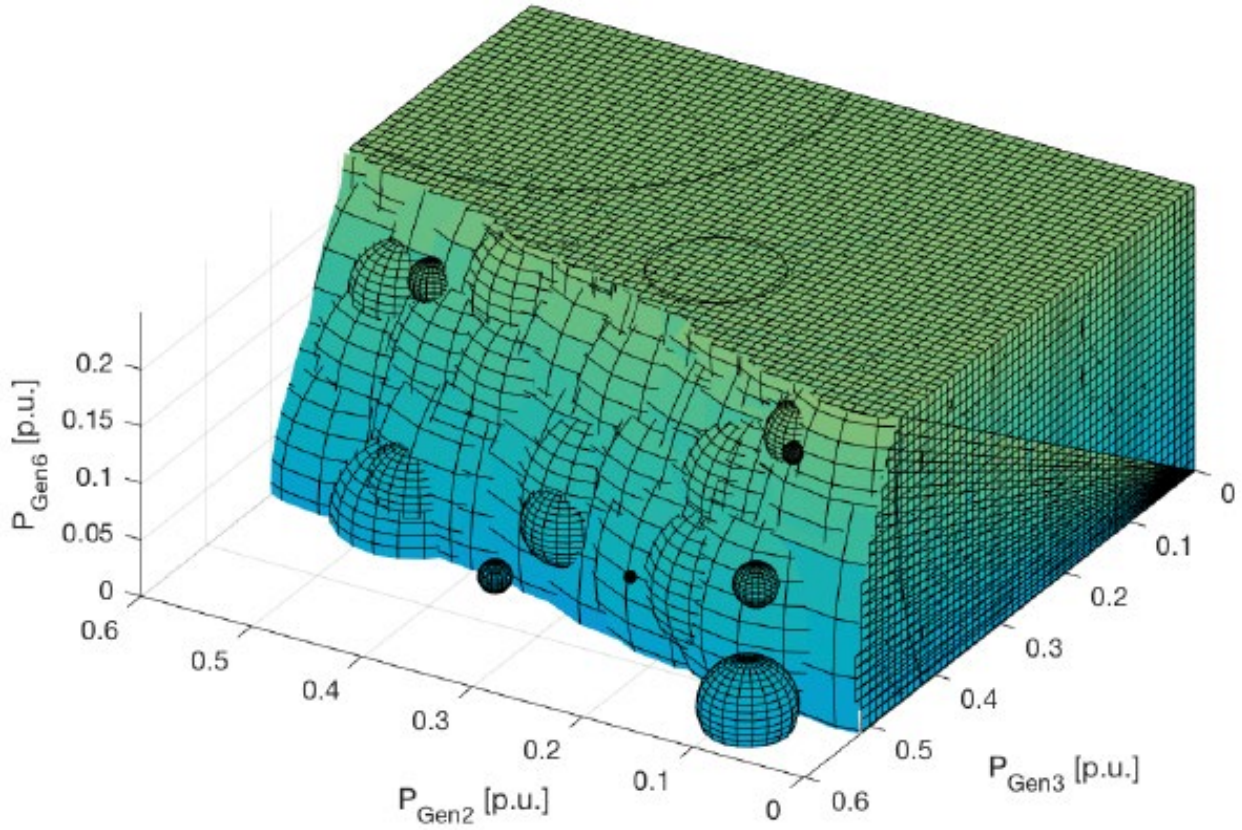


- Certificate: if point infeasible for semidefinite relaxation → infeasible for the original problem
- If infeasible point: find minimum radius to feasibility

Convex relaxations to discard infeasible regions



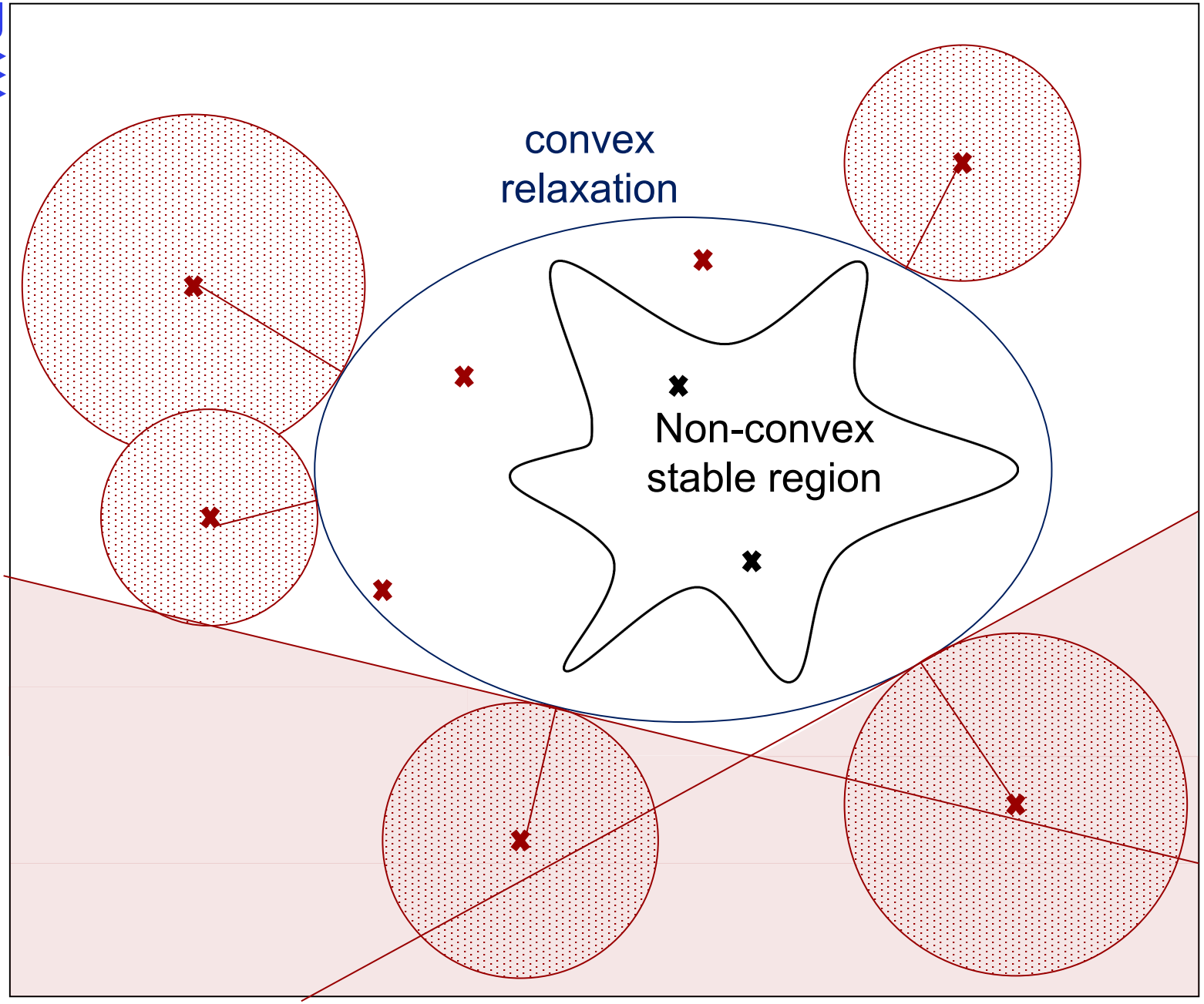
- Certificate: if point infeasible for semidefinite relaxation → infeasible for the original problem
- If infeasible point: find minimum radius to feasibility
- Discard all points inside the (hyper)sphere



- 3D projection of hyperspheres
- IEEE 14-bus system
- Rapidly discarding (=classifying) large chunks of the search space as infeasible to focus on the boundary

F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems". IEEE Trans. Power Systems, vol. 35, no. 1, pp. 30-41, Jan. 2020.. https://www.arxiv.org/abs/1806.0107.pdf

Convex relaxations to discard infeasible regions



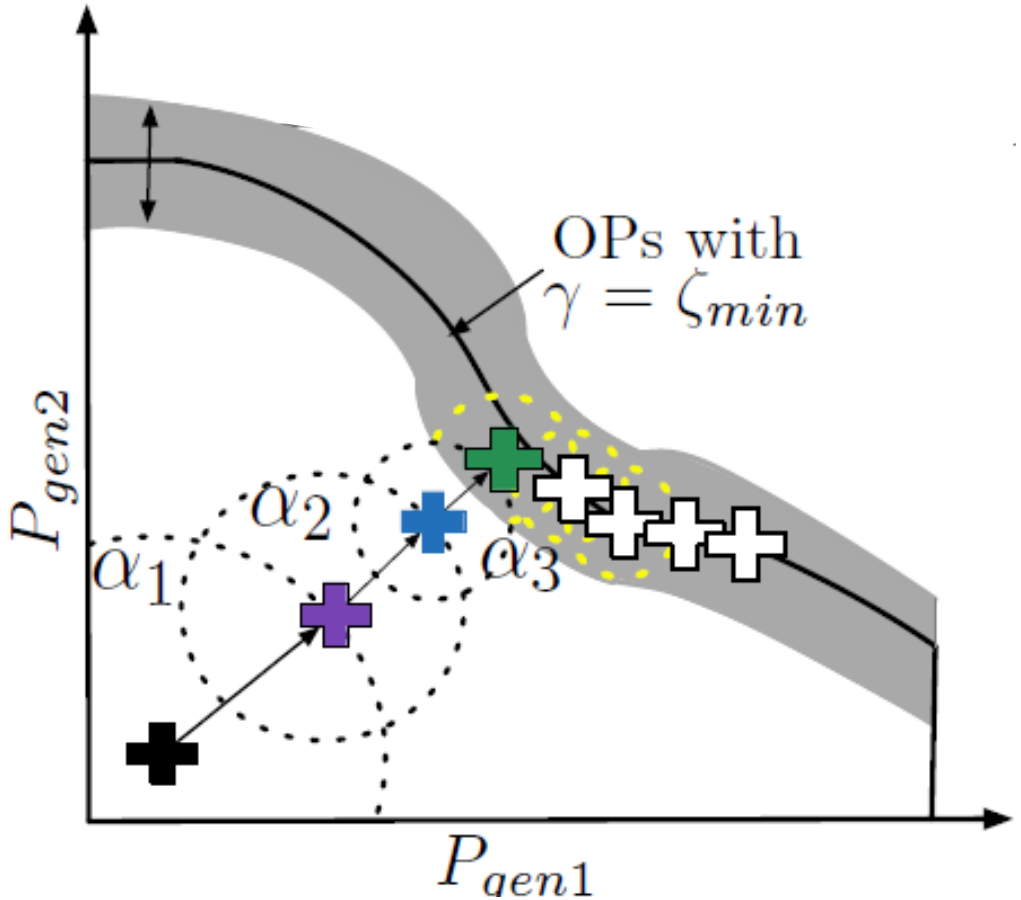
- Extension of this work to hyperplanes

- A. Venzke, D.K. Molzahn, S. Chatzivasileiadis, **Efficient Creation of Datasets for Data-Driven Power System Applications**. PSCC 2020. https://arxiv.org/pdf/1910.01794.pdf

Directed Walks

- "Directed walks": steepest-descent based algorithm to explore the remaining search space, focusing on the area around the security boundary

1. Variable step-size
2. Parallel computation
3. Full N-1 contingency check



Results

	Points close to the security boundary (within distance γ)	
	IEEE 14-bus	NESTA 162-bus
Brute Force	100% of points in **556.0 min**	*intractable*
Importance Sampling	100% of points in **37.0 min**	**901 points** in 35.7 hours
Proposed Method	100% of points in **3.8 min**	**183'295 points** in 37.1 hours

F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems". ". IEEE Trans. Power Systems, vol. 35, no. 1, pp. 30-41, Jan. 2020. https://www.arxiv.org/abs/1806.0107.pdf

Revisited in 2025: We compared 3 approaches

1. Latin Hypercube Sampling (LHC) = uniform sampling across the input domain
2. Importance Sampling
3. (Our) Proposed Method

For 39-bus and 162-bus systems

- Open-SourceToolbox: https://github.com/bastiengiraud/DSA-learn

GitHub

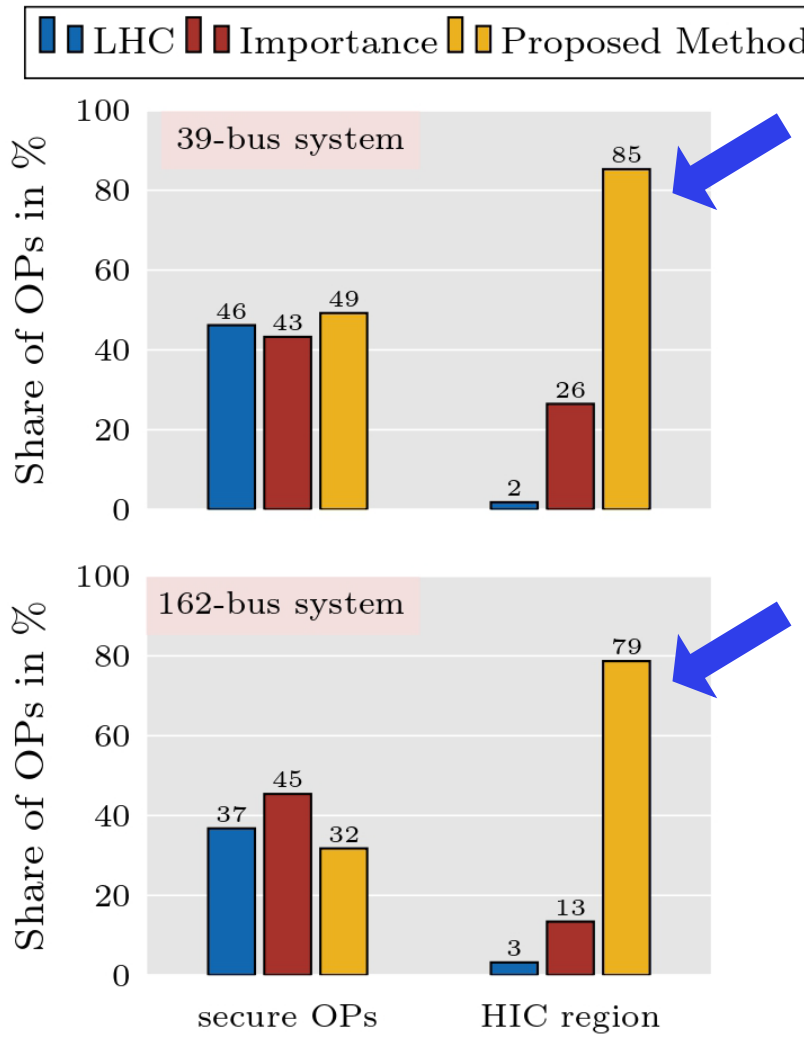
Implemented in Julia using Powermodels.jl, PowerSystems.jl and PowerSimulationsDynamics.jl

B. Giraud , L. Charles, A. M. Nakiganda, J. Vorwerk, S. Chatzivasileiadis, A Dataset Generation Toolbox for Dynamic Security Assessment: On the Role of the Security Boundary, Sustainable Energy, Grids, and Networks, Elsevier, 2025, https://arxiv.org/pdf/2501.09513

Share of OPs in HIC region

HIC = High Information Content

OP = Operating Point

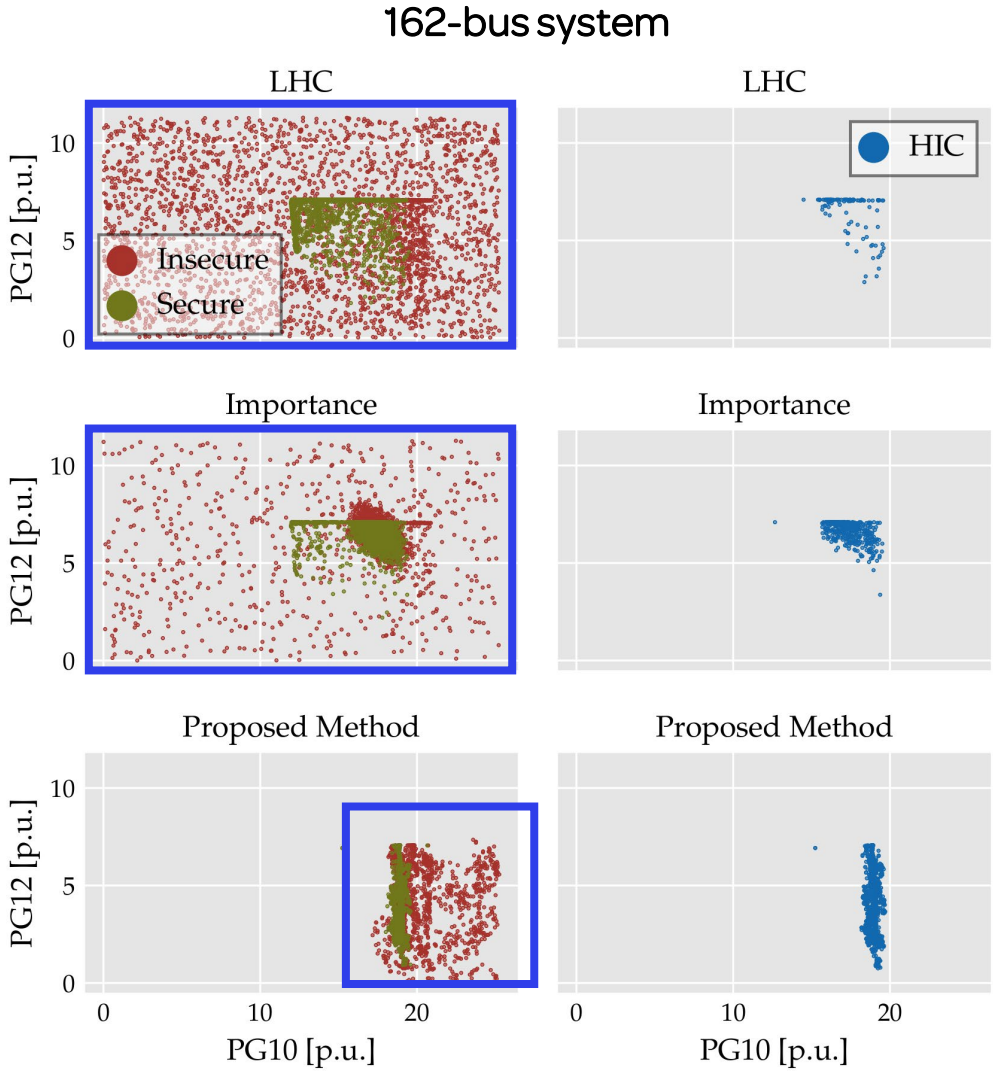


- All methods can produce balanced datasets
- Our proposed method has a much higher share of OPs in the HIC region

Distribution of OPs

HIC = High Information Content

OP = Operating Point



- Proposed algorithm drastically reduces the search space:
  - LHC vs Proposed: over **2'500x** input space reduction

- Security boundary not fully defined with all algorithms → shows how complex the task is

Computation Time

HIC = High Information Content

OP = Operating Point

TABLE III
COMPUTATIONAL COST COMPARISON FOR THE 39-BUS SYSTEM.

Method	Time / 1k HIC OPs
LHC	9.9 h
Importance	0.4 h
Proposed Method	0.2 h

~50x faster

TABLE IV
COMPUTATIONAL COST COMPARISON FOR THE 162-BUS SYSTEM.

Method	Time / 1k HIC OPs
LHC	36.9 h
Importance	8.0 h
Proposed Method	0.5 h

~74x faster

- Proposed method more than **50x faster** than naive sampling approach to sample 1k OPs in the HIC region.

Question: how important is this high share of OPs in the HIC region?

We trained Decision Trees

- We trained a Decision Tree (DT) with each dataset and tested it on all 3.
  - And we also tested the DT on test dataset which collected samples right around the HIC region (boundary)

F1-score for the 39-bus system

Dataset we trained with \ Dataset we tested for	LHC	Importance	Proposed Method	Boundary
LHC	0.98	0.62	0.78	0.67
Importance	0.95	0.92	0.67	0.64
Proposed Method	0.95	0.91	0.92	0.77

F1-score: Harmonic mean of Precision and Recall

$$\mathrm{F1} = \frac{Precision \cdot Recall}{Precision + Recall}$$

Recall: True Positive Rate $\frac{TP}{TP+FN}$

Precision: Positive Predictive Value $\frac{TP}{TP+FP}$

We trained Decision Trees

- We trained a Decision Tree (DT) with each dataset and tested it on all 3.
  - And we also tested the DT on test dataset which collected samples right around the HIC region (boundary)

F1-score for the 39-bus system

Dataset we trained with \ Dataset we tested for	LHC	Importance	Proposed Method	Boundary
LHC	0.98	0.62	0.78	0.67
Importance	0.95	0.92	0.67	0.64
Proposed Method	0.95	0.91	0.92	0.77

- Each DT performs best on the test dataset it was trained for

F1-score: Harmonic mean of Precision and Recall

$$\mathrm{F1} = \frac{Precision \cdot Recall}{Precision + Recall}$$

Recall: True Positive Rate $\frac{TP}{TP+FN}$

Precision: Positive Predictive Value $\frac{TP}{TP+FP}$

We trained Decision Trees

- We trained a Decision Tree (DT) with each dataset and tested it on all 3.
  - And we also tested the DT on test dataset which collected samples right around the HIC region (boundary)

F1-score for the 39-bus system

Dataset we trained with \ Dataset we tested for	LHC	Importance	Proposed Method	Boundary
LHC	0.98	0.62	0.78	0.67
Importance	0.95	0.92	0.67	0.64
Proposed Method	0.95	0.91	0.92	0.77

- Each DT performs best on the test dataset it was trained for
- On the boundary, our proposed method outperforms the rest
- **Overall, our proposed method has the best overall performance**

F1-score: Harmonic mean of Precision and Recall

$$\mathrm{F1} = \frac{Precision \cdot Recall}{Precision + Recall}$$

Recall: True Positive Rate $\frac{TP}{TP+FN}$

Precision: Positive Predictive Value $\frac{TP}{TP+FP}$

A Balanced Dataset significantly improves the classifier performance

- 162-bus
- Trained only with our proposed method
  - Sampled a balanced and an unbalanced training dataset
- Tested on all test datasets

TABLE V
162-BUS SYSTEM F1-SCORES WITH AN UNBALANCED AND A BALANCED DATASET FOR THE PROPOSED METHOD.

Training \ Testing	LHC	Importance	Proposed Method	Boundary
Unbalanced	0.90	0.69	0.64	0.59
Balanced	0.97	0.87	0.88	0.85

(Repeat) Takeaway #3:
Neural networks (or Decision Trees) for classification: you need a balanced training database → similar number of safe and unsafe points

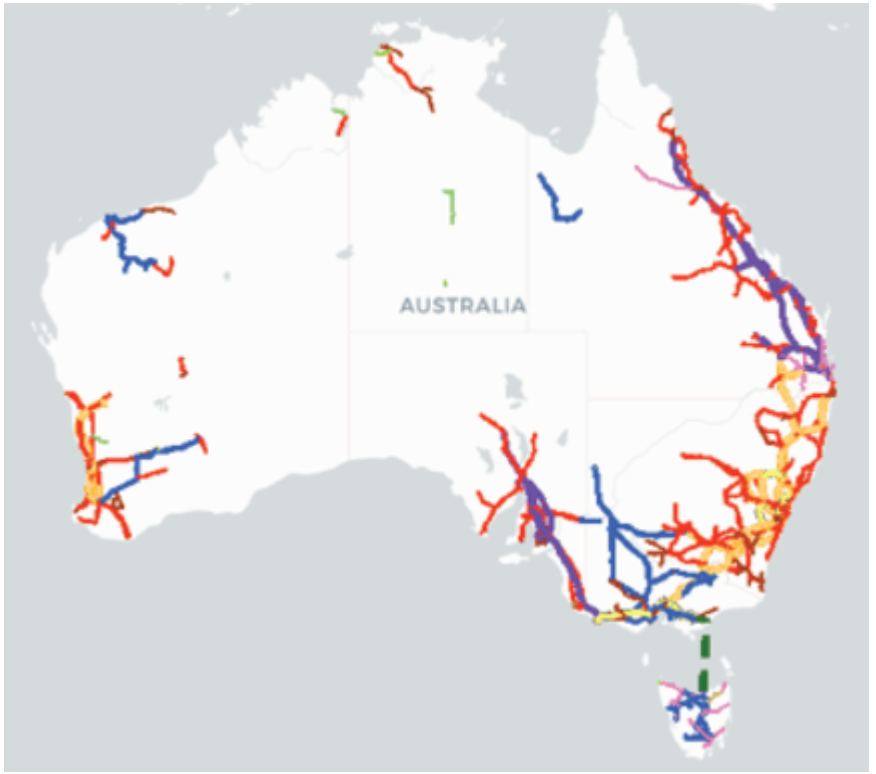
Takeaway #6

Creating high-quality training databases is extremely complex and an open research topic. We need to go beyond purely statistical methods and exploit the underlying physics during sampling.

Physics-Informed Neural Networks for Power System Dynamics

What is the challenge?

- Assume we operate the **Australian Grid** and need to eliminate the blackout risk for the next day.
- We need simulations to assess the risk and devise mitigation strategies.
- **Simulating 20 seconds of the dynamic behavior** of the Australian Grid requires **12 minutes** with a current state-of-the-art tool.
- In a system of hundreds of nodes, there are 1,000s of potential contingencies, and 100s of operating points that appear in a day.
- Suppose we check just the 100 most critical disturbances for 5 hopefully representative operating points. **This requires non-stop simulations for 4 days.**



- Performing such a task every day is impossible.
- **Our goal bring this time down from 4 days to 1 hour (100x speedup)**

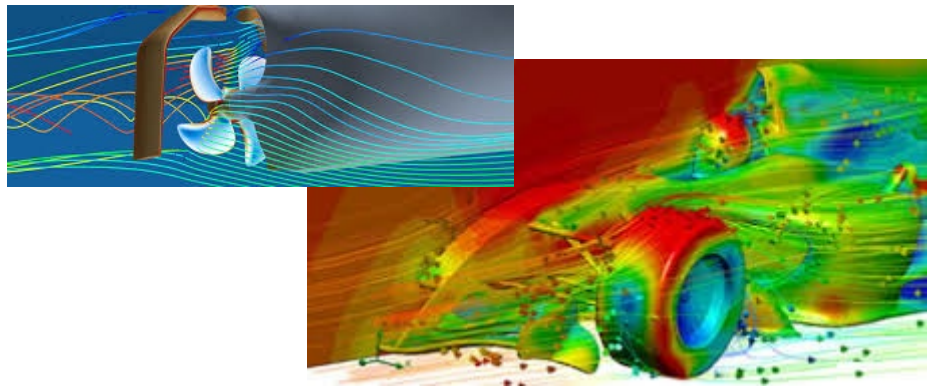
Physics-Informed Neural Networks (PINNs)

- **Why can Neural Networks be faster than conventional simulation tools?**
  - Conventional tools need to run iterative methods to approximate the solution of differential equations
  - For Neural Networks, it is a forward matrix multiplication (as long as they are accurate enough)

- **What is the benefit of PINNs over standard NNs?**
  - **PINNs do not need large amounts of training data.** They learn from the physical models included in training.
  - No need to spend (a lot of) time on generating data or depend on incomplete data

10x-100x-1'000x faster solution, depending on the application

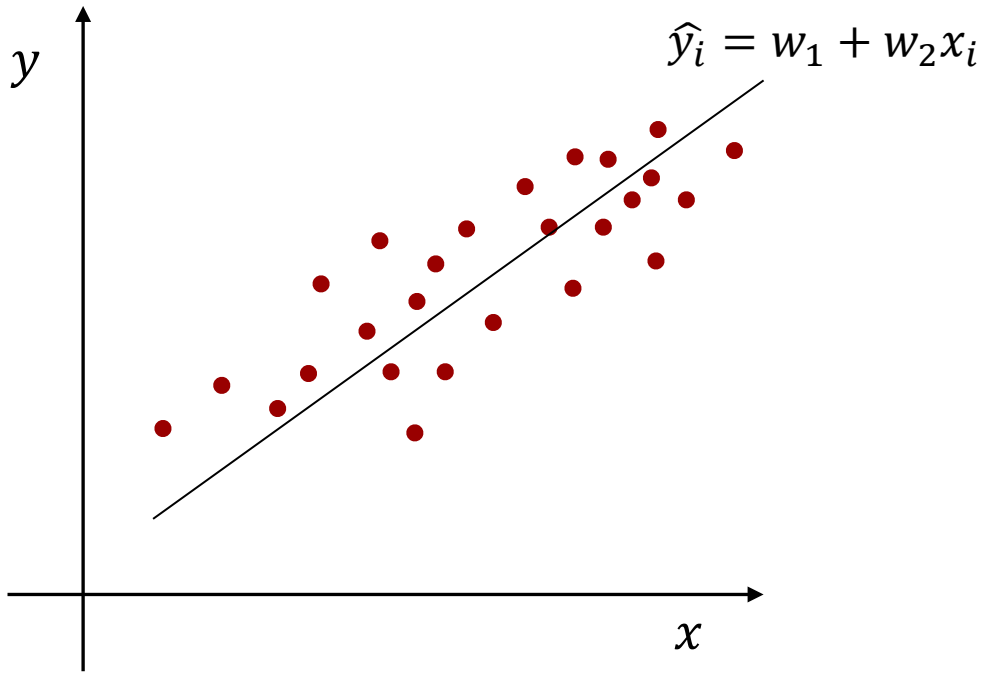
Seem to be achieving significant speedups for partial differential equations (e.g. computational fluid dynamics)



Neural Networks: An advanced form of non-linear regression

y_i: actual/correct value

$\hat{y}_i$: estimated value



Loss function: Estimate best w_1, w_2 to fit the training data

$$\min_{w_1, w_2} \quad \|y_i - \hat{y}_i\|$$

s.t.

$$\hat{y}_i = w_1 + w_2 x_i \quad \forall i$$

Traditional training of neural networks required no information about the underlying physical model. Just data!

Physics Informed Neural Networks

- Automatic differentiation: derivatives of the neural network output with respect to the input can be computed during the training procedure
- A differential-algebraic model of a physical system can be included in the neural network training*
- Neural networks can now exploit knowledge of the actual physical system
- Machine learning platforms (e.g. Pytorch, Tensorflow) enable these capabilities

*M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-Informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", Journal of Computational Physics, vol.378, pp. 686-707, 2019

Physics-Informed Neural Networks for Power Systems

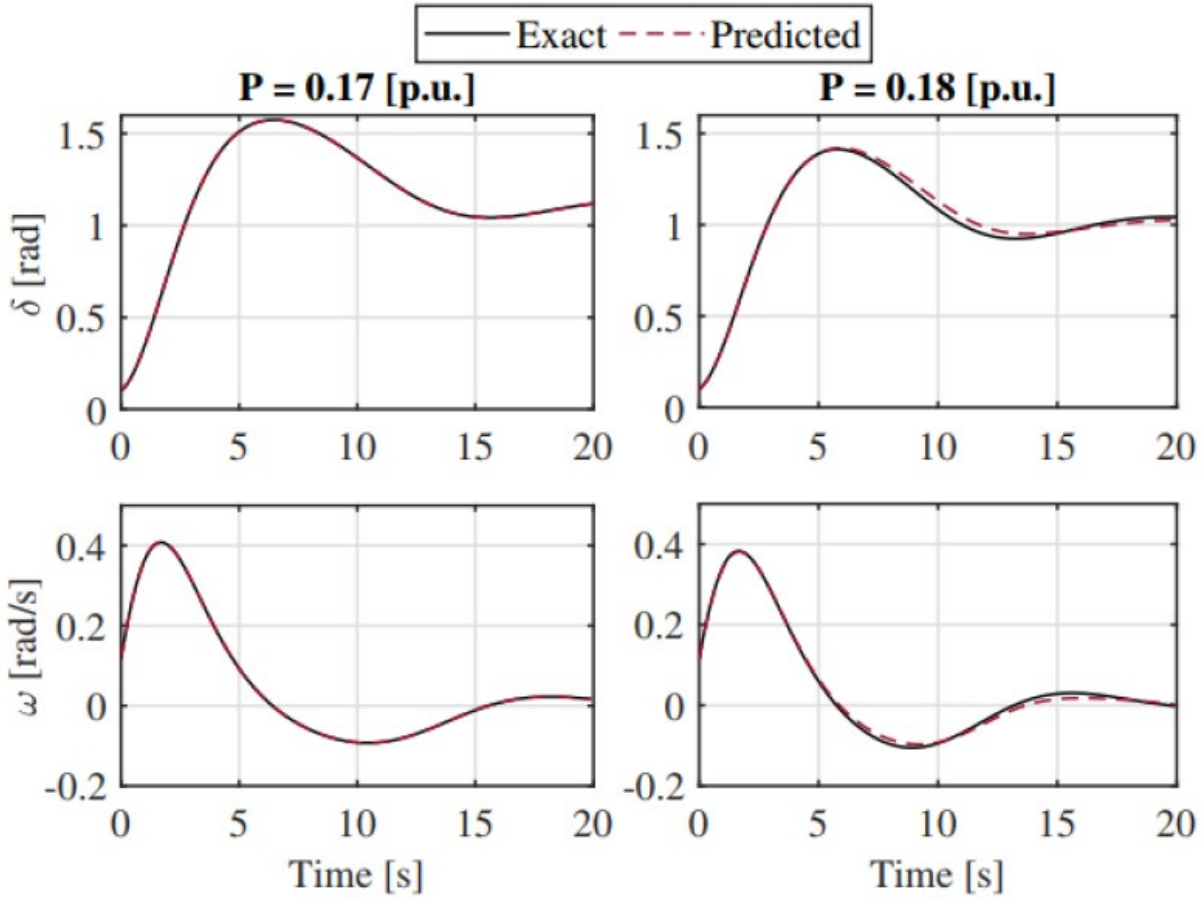
"Original"
Loss function

$$\min_{\mathbf{W},\mathbf{b}} \quad \frac{1}{|N_\delta|}\sum_{i\in N_\delta}|\hat{\delta}-\delta^i|^2+\frac{1}{|N_f|}\sum_{i\in N_f}|f(\hat{\delta})|^2 \tag{6a}$$

$$s.t. \qquad \hat{\delta}=NN(t,P_m,\mathbf{W},\mathbf{b}) \tag{6b}$$

$$\dot{\hat{\delta}}=\frac{\partial\hat{\delta}}{\partial t}, \qquad \ddot{\hat{\delta}}=\frac{\partial\dot{\hat{\delta}}}{\partial t} \tag{6c}$$

$$f(\hat{\delta})=M\ddot{\hat{\delta}}+D\dot{\hat{\delta}}+A\sin\hat{\delta}-P_m \tag{6d}$$



G. S. Misyris, A. Venzke, S. Chatzivasileiadis, **Physics-Informed Neural Networks for Power Systems**. Presented at the **Best Paper Session** of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

Physics-Informed Neural Networks for Power Systems

"Original" Loss function

"Physics-Informed" term

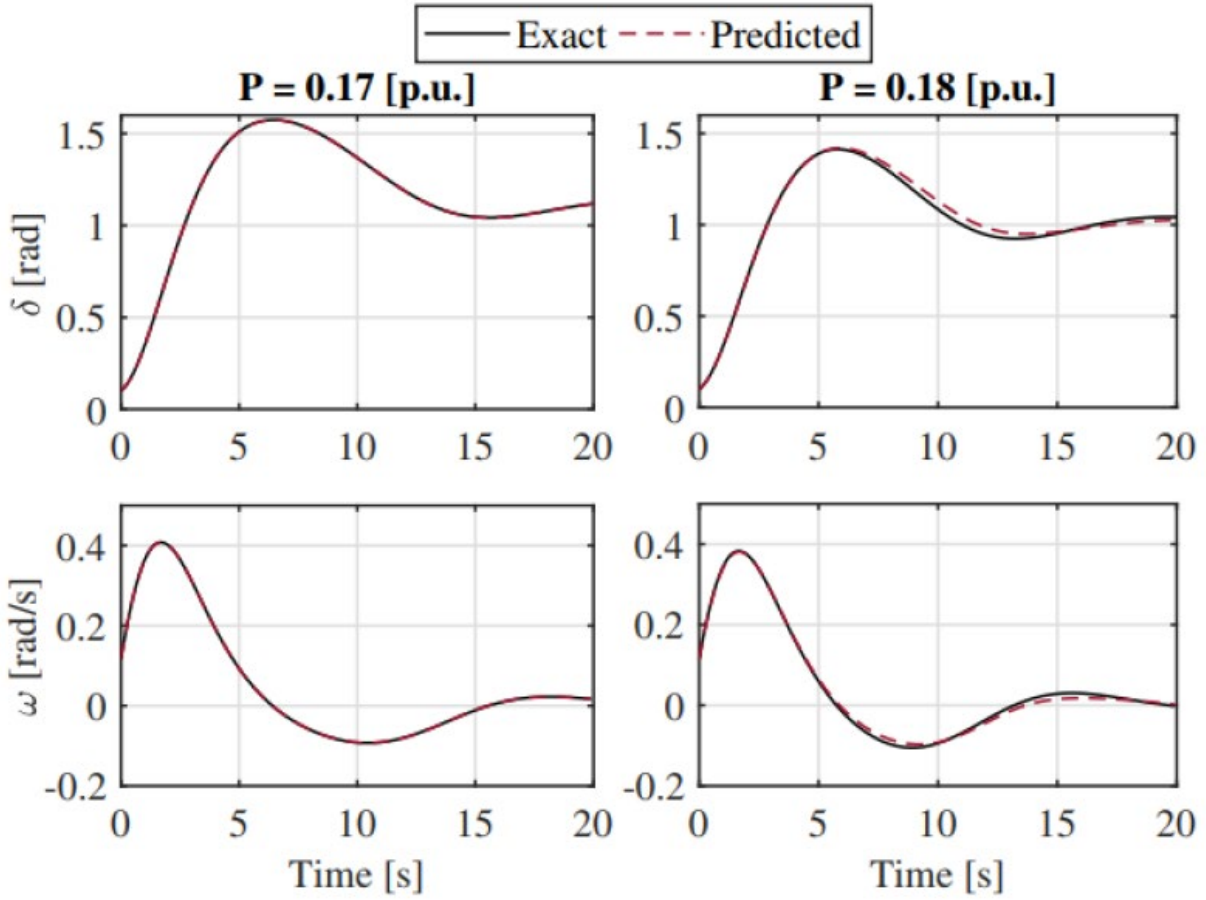
$$\min_{\mathbf{W},\mathbf{b}} \quad \frac{1}{|N_\delta|}\sum_{i\in N_\delta}|\hat{\delta}-\delta^i|^2+\frac{1}{|N_f|}\sum_{i\in N_f}|f(\hat{\delta})|^2 \tag{6a}$$

$$s.t. \qquad \hat{\delta}=NN(t,P_m,\mathbf{W},\mathbf{b}) \tag{6b}$$

$$\dot{\hat{\delta}}=\frac{\partial\hat{\delta}}{\partial t}, \qquad \ddot{\hat{\delta}}=\frac{\partial\dot{\hat{\delta}}}{\partial t} \tag{6c}$$

$$f(\hat{\delta})=M\ddot{\hat{\delta}}+D\dot{\hat{\delta}}+A\sin\hat{\delta}-P_m \tag{6d}$$

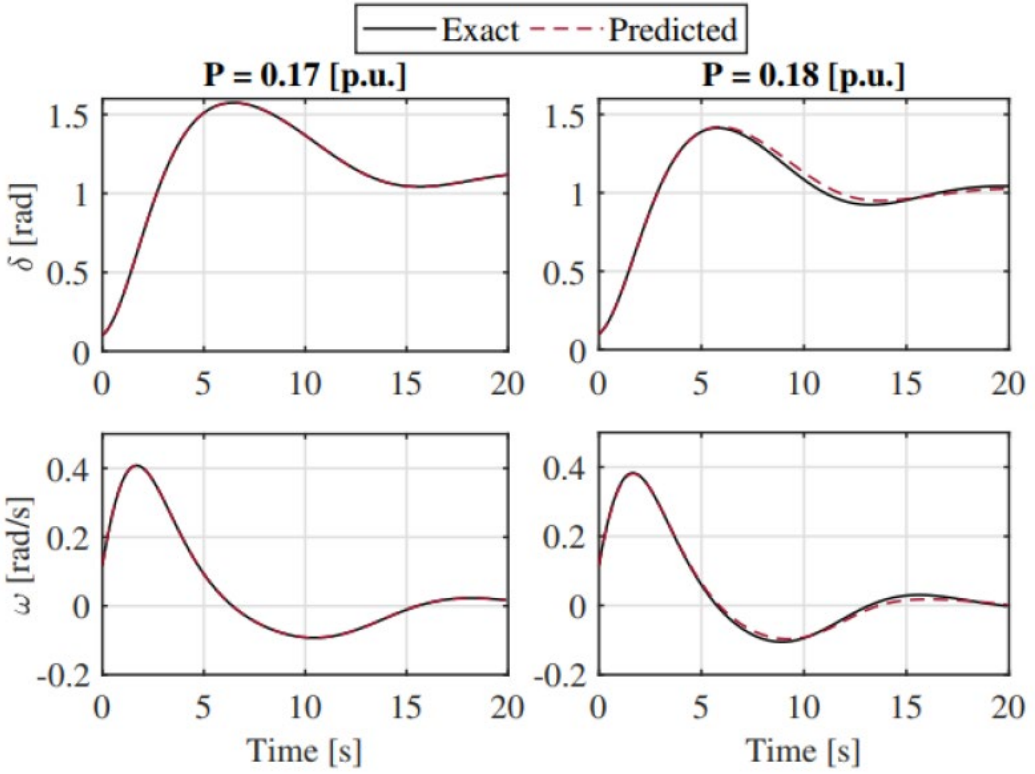
Swing equation



G. S. Misyris, A. Venzke, S. Chatzivasileiadis, **Physics-Informed Neural Networks for Power Systems**. Presented at the **Best Paper Session** of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

Physics-Informed Neural Networks for Power Systems

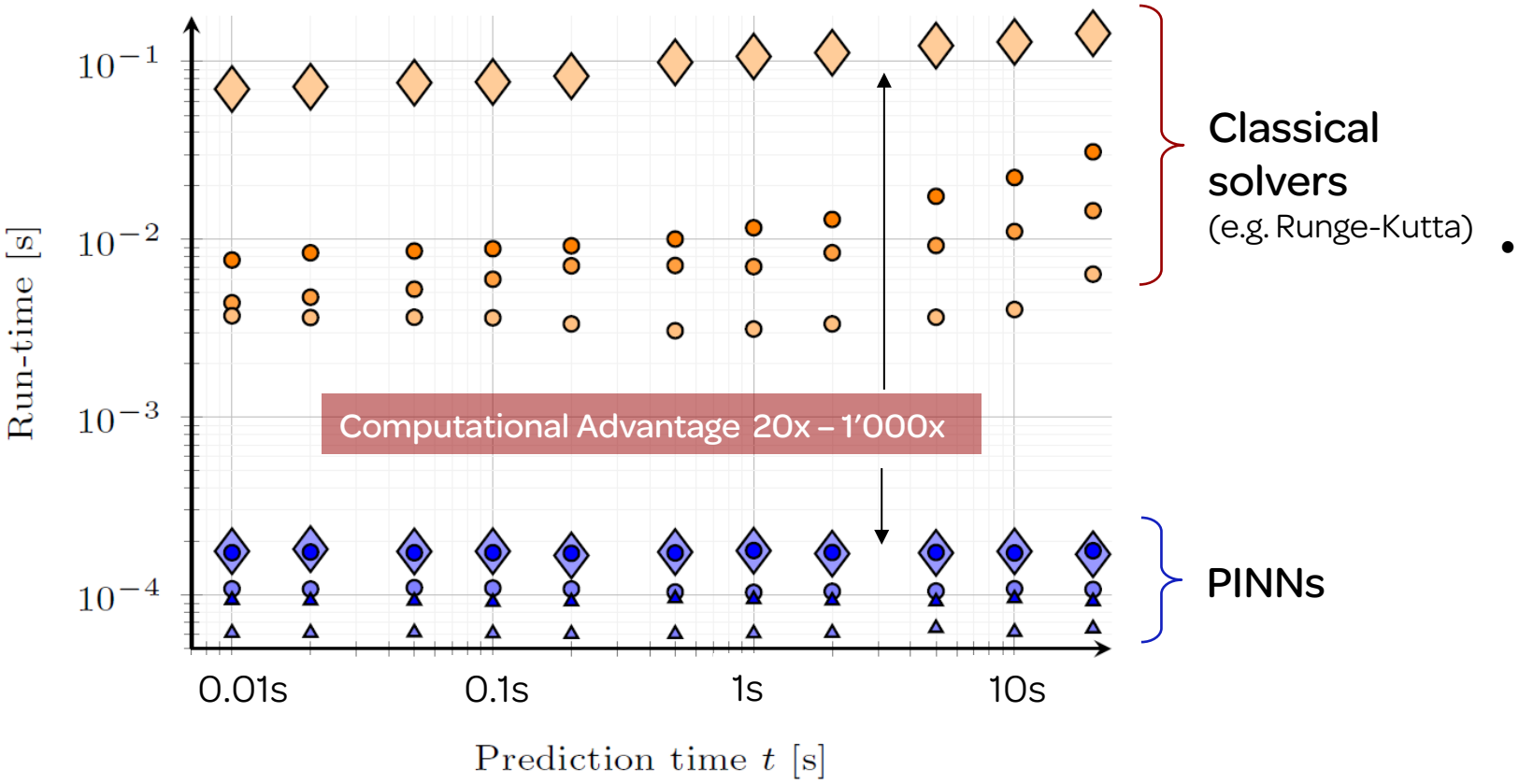
- Physics-Informed Neural Networks (PINN) could potentially replace solvers for systems of differential-algebraic equations in the long-term
  - **Probable power system application: Extremely fast screening of critical contingencies**
- In our example: PINN 87 times faster than ODE solver
- Can **directly estimate** the rotor angle at **any** time instant



Code is available on GitHub: https://github.com/jbesty

G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the Best Paper Session of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf

Computation time: Classical numerical solvers vs. Physics-Informed NNs



Results from 11-bus and 39-bus

- **Physics-Informed Neural Networks** can determine the outputs on average **100x faster than classical numerical solvers**
  - The further ahead we look in time, the larger the computational advantage is

J. Stiasny, S. Chatzivasileiadis, Physics-Informed Neural Networks for Time-Domain Simulations: Accuracy, Computational Cost, and Flexibility https://arxiv.org/abs/2303.08994 [code]

But, there is a trade-off. The further we look ahead in time, the more difficult the learning task becomes

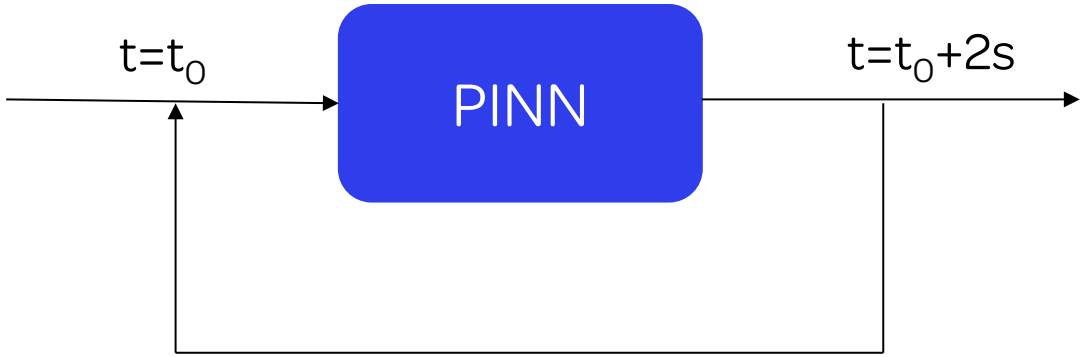
Learning takes longer

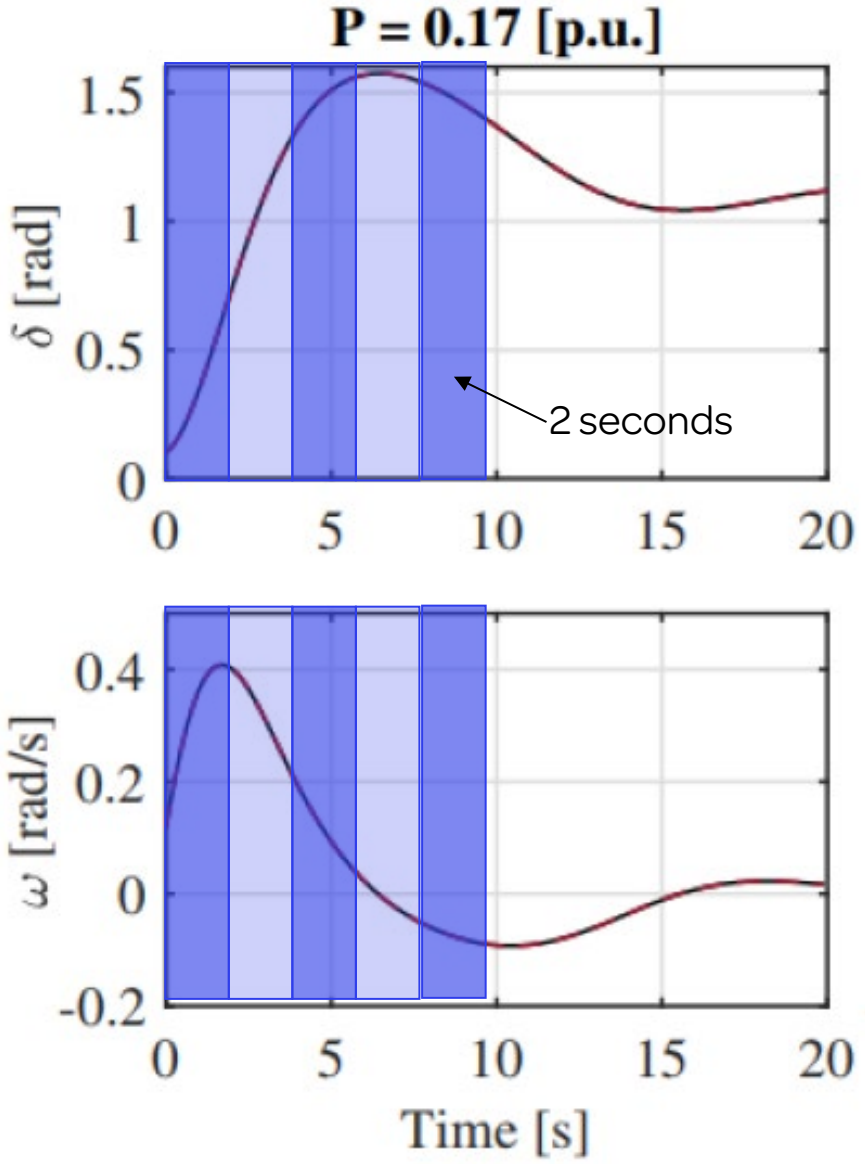
PINN accuracy drops

What shall we do?

How can we reduce training time and improve performance?

1. Train for a shorter time period but for a wide range of initial conditions
2. Use the PINN in a recurrent fashion





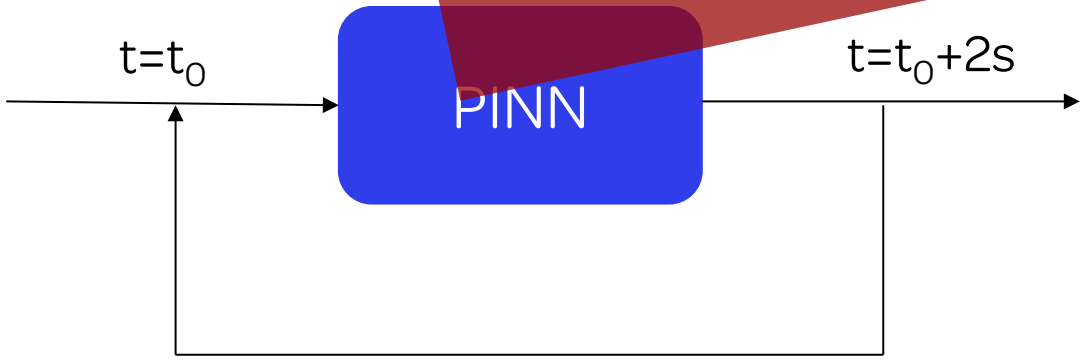
How can we reduce training time and improve performance?

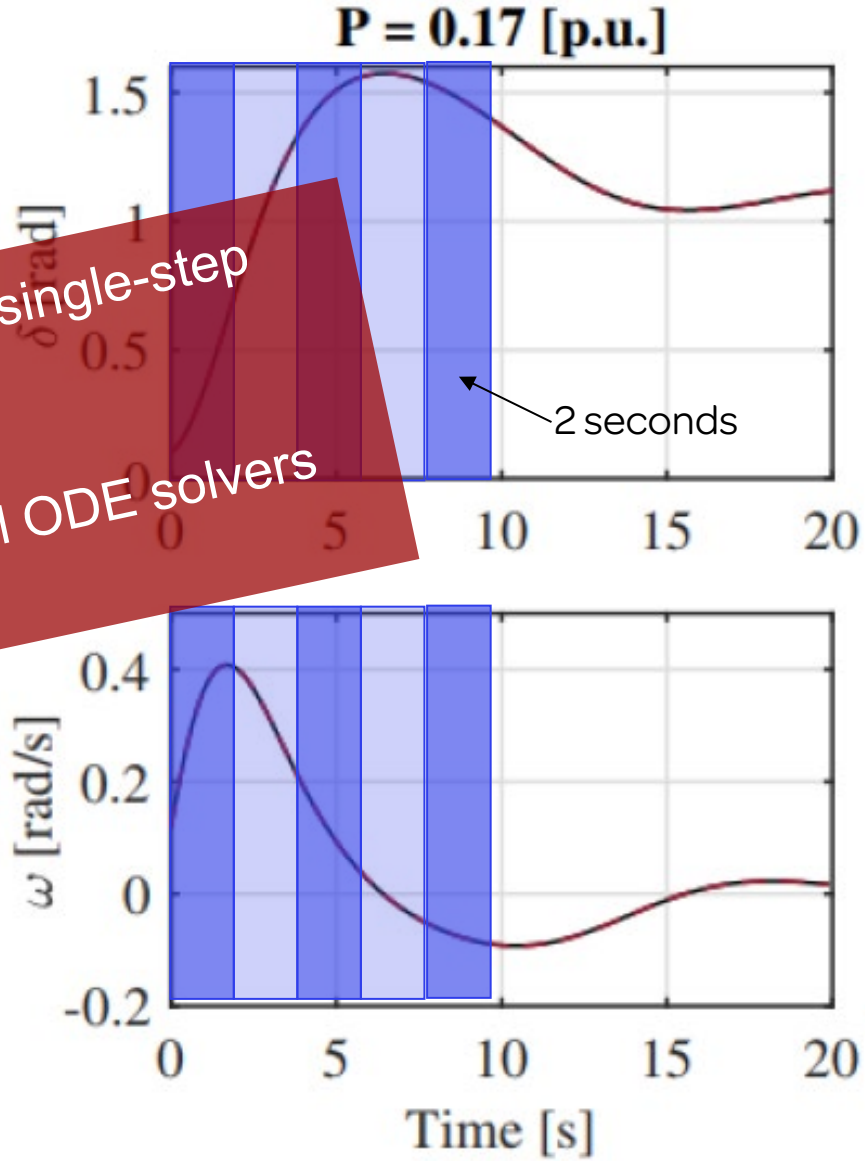
1. Train for a shorter time period but for a wide range of initial conditions
2. Use the PINN in a recurrent fashion

Is the recurrent PINN slower than the single-step PINN?

Yes.

But still 100x faster than conventional ODE solvers (for larger problems)

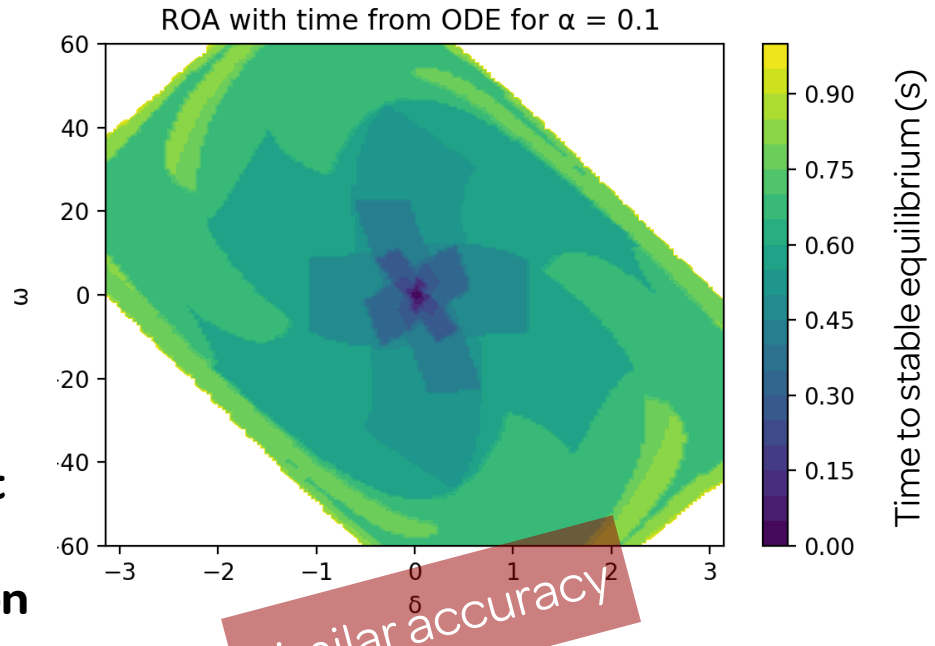




Simulations for Wind Farms:

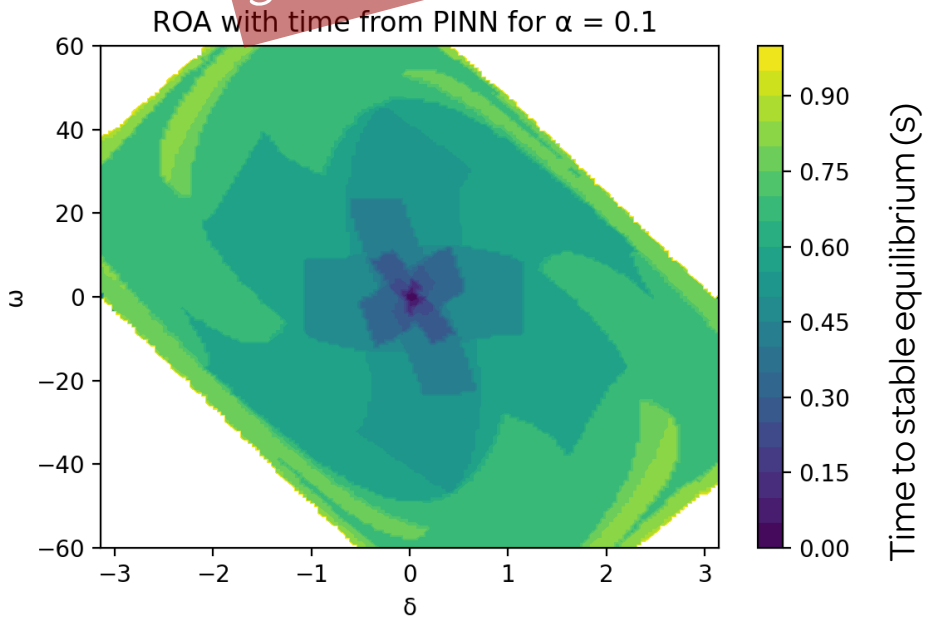
Estimating the Region of Attraction of a Wind Farm Controller

Electro-magnetic Transient (EMT) Simulation



Similar accuracy

PINN



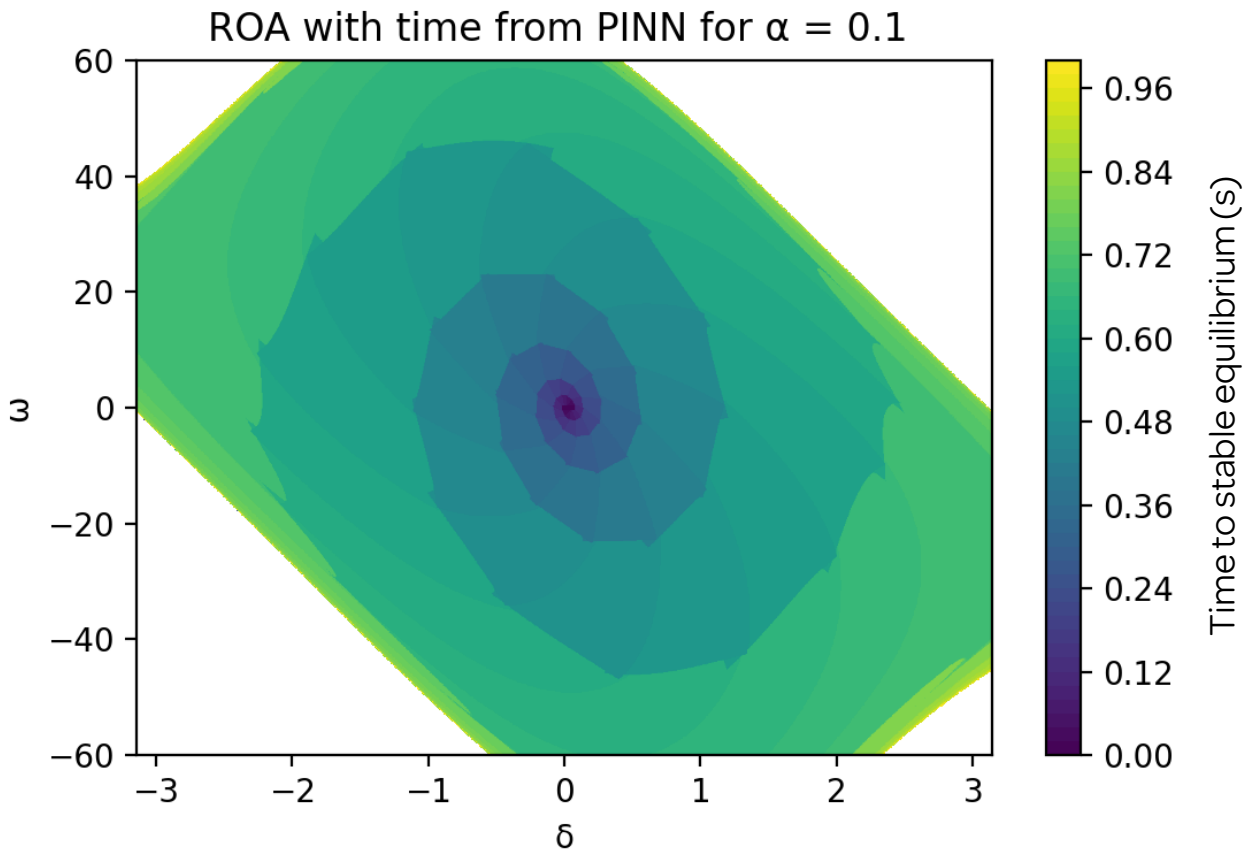
- Collaboration with Ørsted
  - Estimating the region of attraction of controllers is an important part of the wind farm design process
  - Need for **Electromagnetic Transient Simulations**
- Goal: Determine the best set of controller parameters (controller tuning)
- Training PINNs with GPUs
  - collaboration with NVIDIA

R. Nellikkath, A. Venzke, M. K. Bakhshizadeh, I. Murzakhanov, S. Chatzivasileiadis, Physics–Informed Neural Networks for Phase Locked Loop Transient Stability Assessment, PSCC 2024 [https://arxiv.org/abs/2303.12116]

Simulations for Wind Farms:

Estimating the Region of Attraction of a Wind Farm Controller

5 million points with PINN



- Evaluation of 5 million points
- EMT: ~2 days @ DTU HPC
- PINNs: 90 minutes for training and 30 minutes for evaluation

} 25x – 100x faster

- Added benefit: once trained, PINN can run on a laptop

R. Nellikkath, A. Venzke, M. K. Bakhshizadeh, I. Murzakhanov, S. Chatzivasileiadis, Physics–Informed Neural Networks for Phase Locked Loop Transient Stability Assessment, PSCC 2024 [https://arxiv.org/abs/2303.12116]

So, what can we do with PINNs?

1. Integrate them in existing simulators and accelerate them
    1. How? Replace the components that slow down the simulation with PINNs.
    2. What are the components that slow down a simulation?
        - The ones with the fastest time constants.
        - Why? If a component changes fast → I need shorter time steps to capture how it changes → i need more time steps to complete a simulation

2. Create a PINN-based Simulator → PINNSim

Most conventional solvers use the Trapezoidal Rule

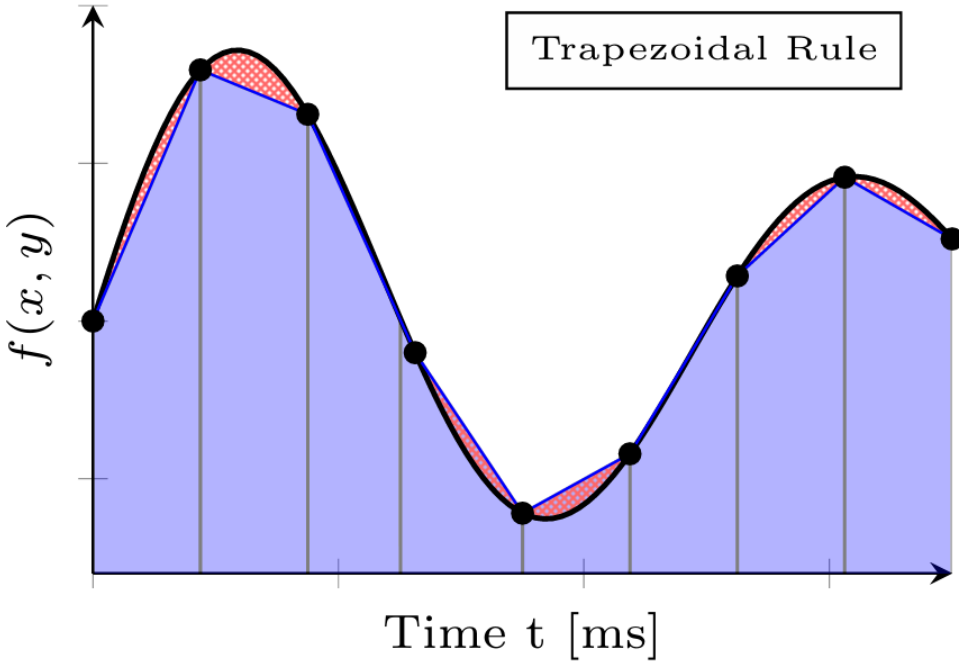
- Implicit numerical method
- Backbone of most power system dynamic simulators
  (PSCAD, EMTP, PowerFactory, and more)

Power System Dynamics ODEs

$$\frac{d}{dt}x = f(x, \bar{V}, \bar{I}, u), \qquad x(t_0) = x_0$$

Trapezoidal Rule

$$x_t = x_{t-\triangle t} + \frac{\triangle t}{2}\left[f(x_t, y_t) + f(x_{t-\triangle t}, y_{t-\triangle t}) \right]$$



Trapezoidal Rule: Quiz

$$x_t = x_{t-\triangle t} + \frac{\triangle t}{2} \left[f(x_t, y_t) + f(x_{t-\triangle t}, y_{t-\triangle t}) \right]$$

When was the trapezoidal rule first used in history?

Trapezoidal Rule: Quiz

$$x_t = x_{t-\triangle t} + \frac{\triangle t}{2} \left[f(x_t, y_t) + f(x_{t-\triangle t}, y_{t-\triangle t}) \right]$$

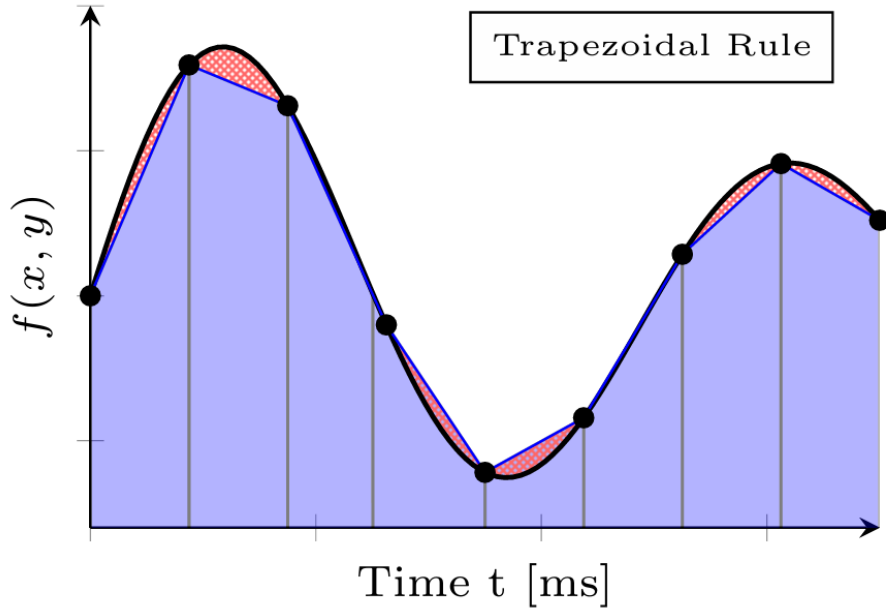
When was the trapezoidal rule first used in history?

50 BCE, Babylon

How can they integrate into simulations?

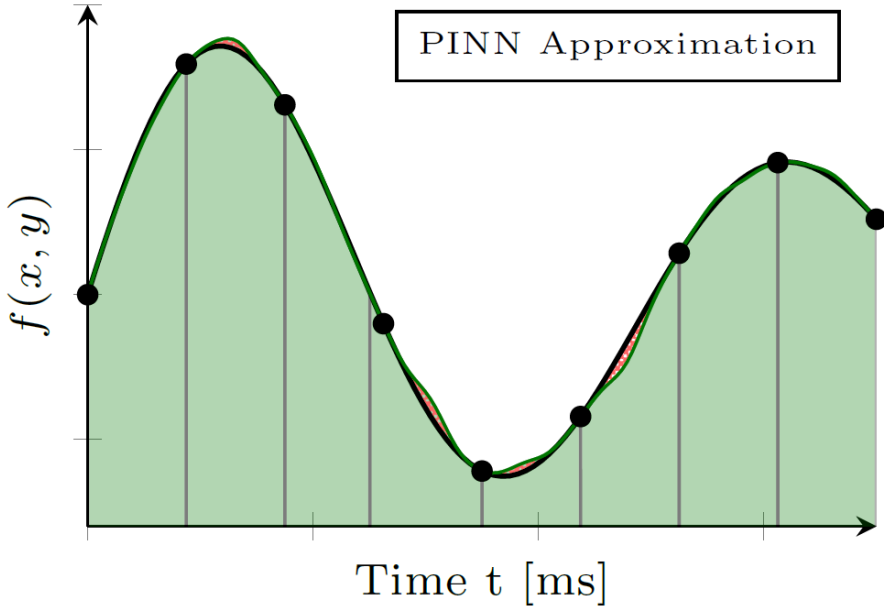
Conventional/Commercial method

$$x_t = x_{t-\Delta t} + \frac{\Delta t}{2}\left[f(x_t, \bar{I}_t) + f(x_{t-\Delta t}, \bar{I}_{t-\Delta t}) \right]$$



PINN Integration

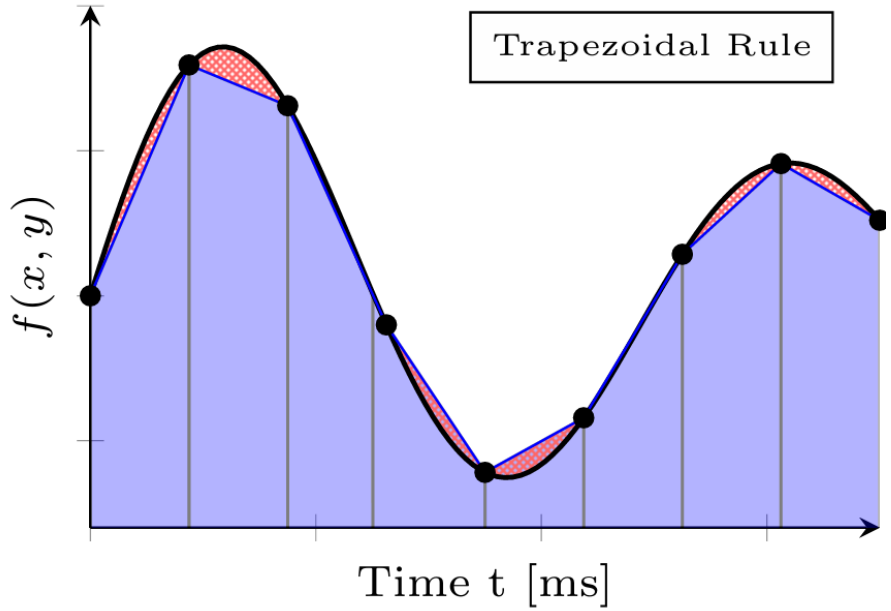
$$x_t = x_{t-\Delta t} + \Delta t \ \mathrm{PINN}(x_{t-\Delta t}, I_{t-\Delta t}, I_t, \Delta t)$$



How can they integrate into simulations?

Conventional/Commercial method

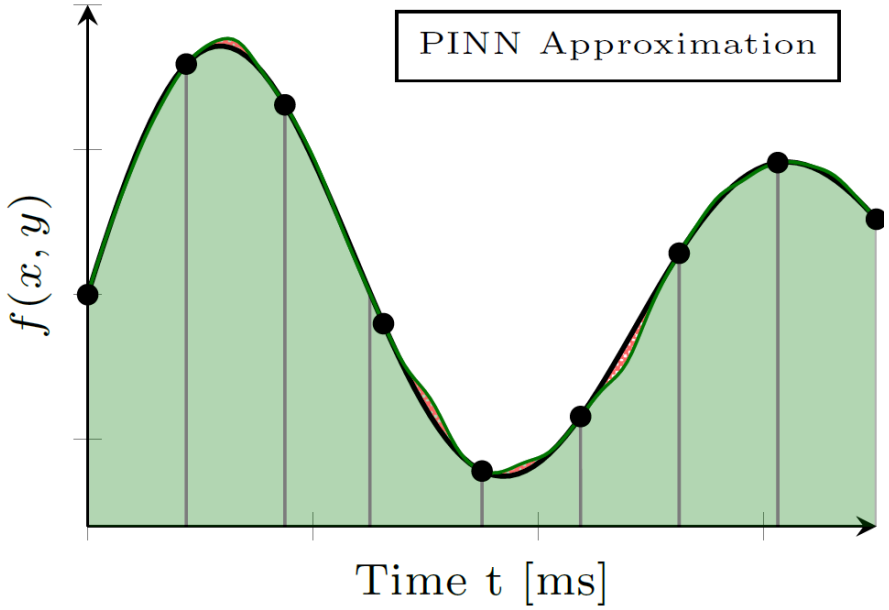
$$x_t = x_{t-\Delta t} + \frac{\Delta t}{2}\left[f(x_t, \bar{I}_t) + f(x_{t-\Delta t}, \bar{I}_{t-\Delta t}) \right]$$



x_t is in both sides of the equation → need iterations to solve it

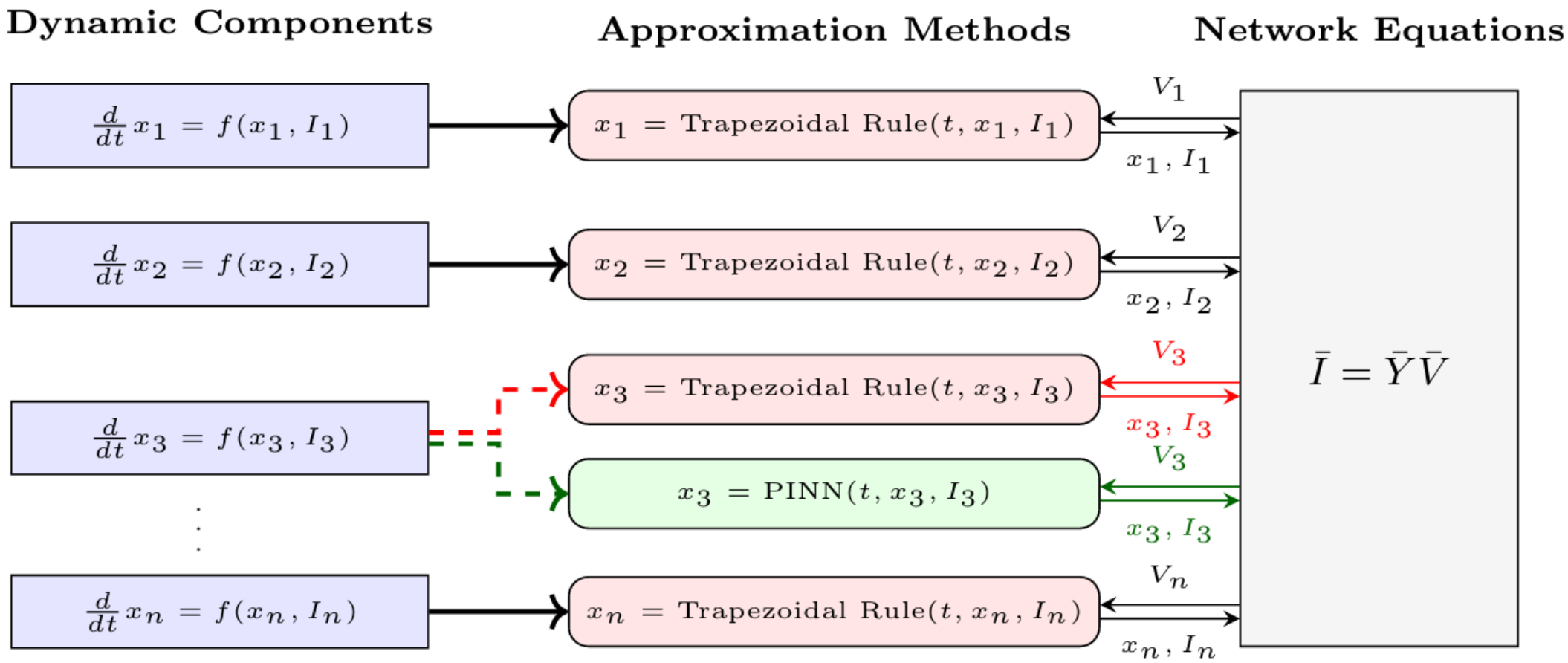
PINN Integration

$$x_t = x_{t-\Delta t} + \Delta t \ \text{PINN}(x_{t-\Delta t}, I_{t-\Delta t}, I_t, \Delta t)$$

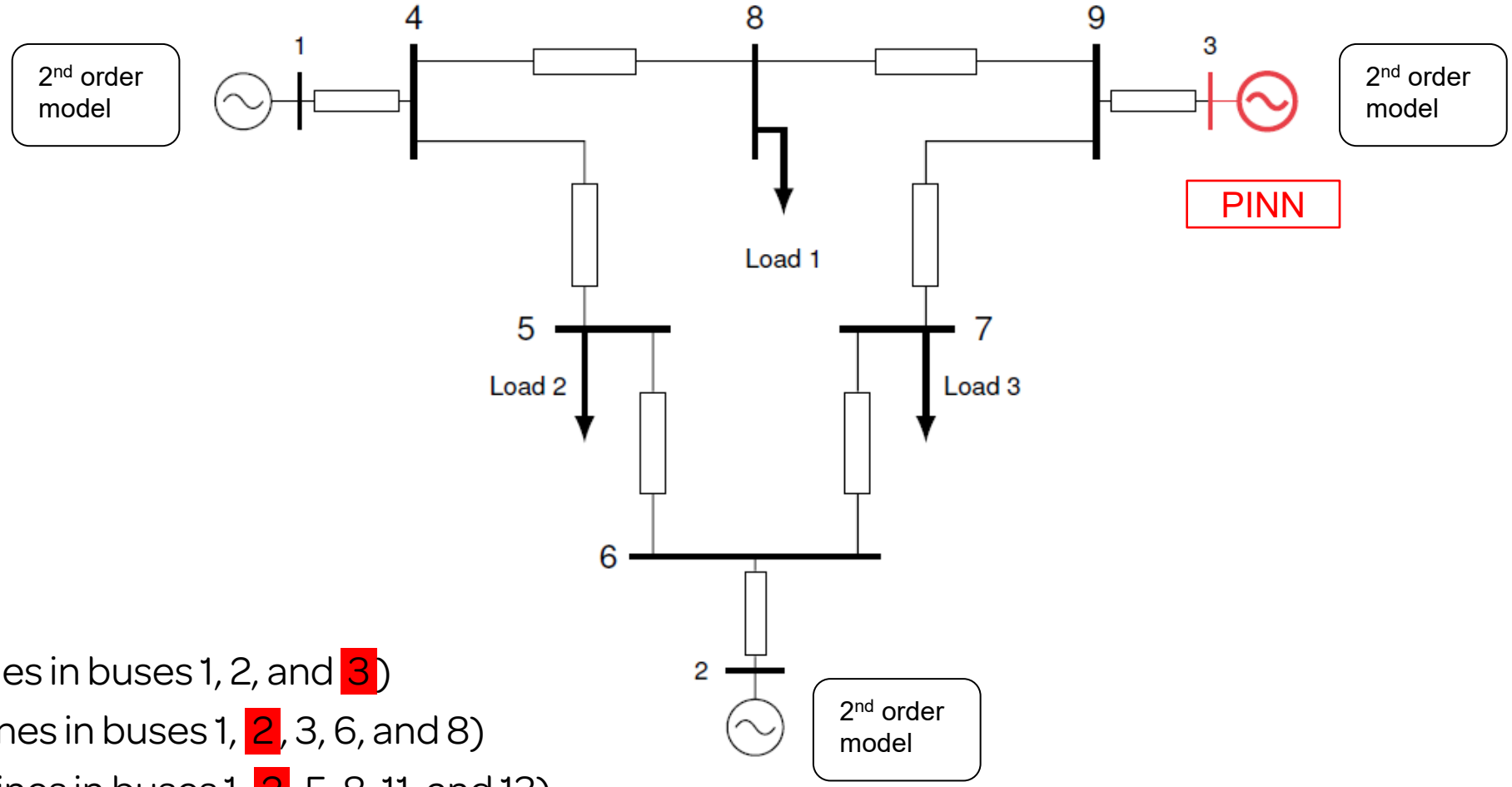


x_t only on the left side → single-step

Plug-and-Play Integration of PINNS



Replacing 1 component with a PINN in dynamic simulations



- 9-Bus System (Machines in buses 1, 2, and 3)
- 14-Bus System (Machines in buses 1, 2, 3, 6, and 8)
- 30-Bus System (Machines in buses 1, 2, 5, 8, 11, and 13)

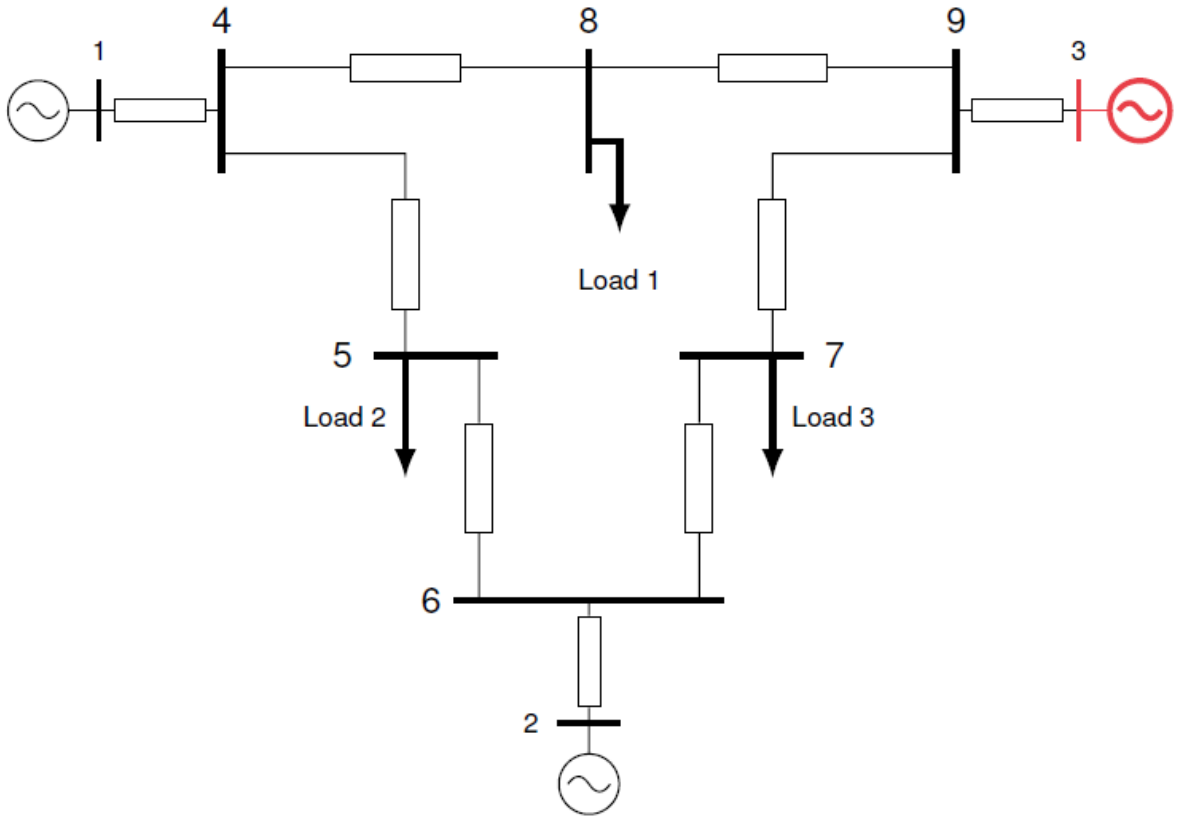
 # marks the machine replaced with a PINN

Integrating PINNs to dynamic simulations

- Including 1 PINN improves the accuracy of the fast component by up to 50%
- Improves overall accuracy on average 1%-18%

System	ω		I_{d-q}	
	ω^{avg}	ω^{*}	I_{d-q}^{avg}	I_{d-q}^{*}
IEEE 9	9.5%	39.9%	17.9%	40.3%
IEEE 14	0.7%	40.5%	8.2%	40.8%
IEEE 30	1.8%	45.6%	8.7%	51.6%

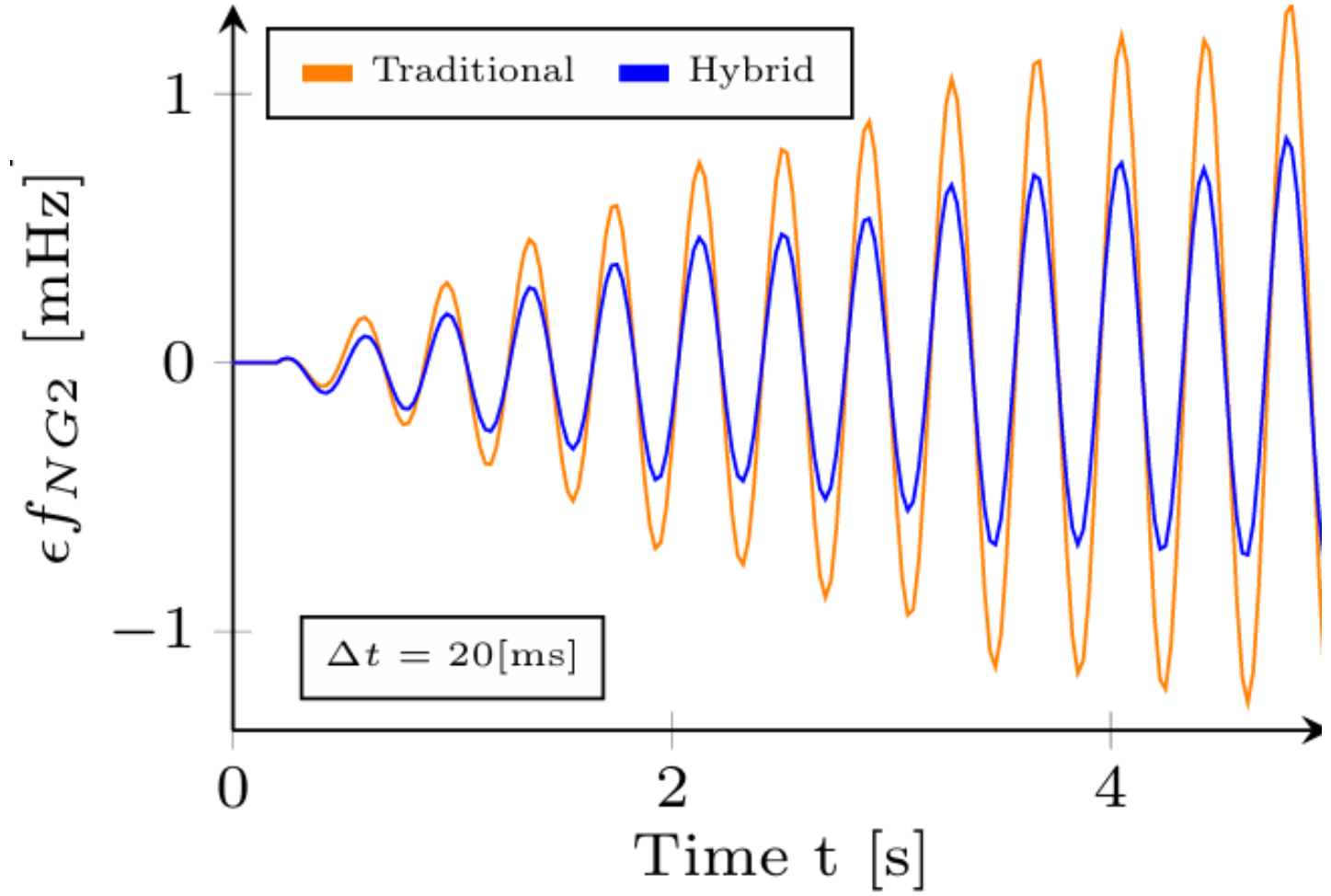
More accurate simulations enable **larger time steps**



I. Ventura-Nadal, J. Stiasny, S. Chatzivasileiadis, Integrating Physics-Informed Neural Networks into Power System Dynamic Simulations, *Electric Power Systems Research*, 2025, https://arxiv.org/pdf/2404.13325

I. Ventura-Nadal, R. Nelikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks in Power System Dynamics: Improving Simulation Accuracy, IEEE Powertech 2025, https://arxiv.org/pdf/2501.17621

IEEE 30-Bus System Simulation

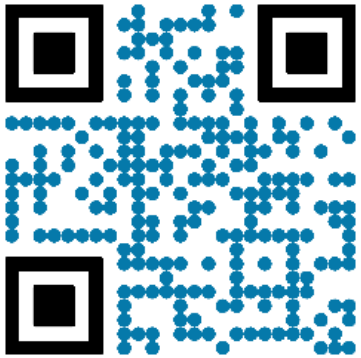


- Traditional:
  - 6 conventionally modelled synchronous machines
- Hybrid:
  - 1 PINN
  - 5 conventionally modelled synchronous machines

Do you want to create your own PINNs?

Open-source Modular Python Toolbox!

GitHub/radiakos/PowerPINN



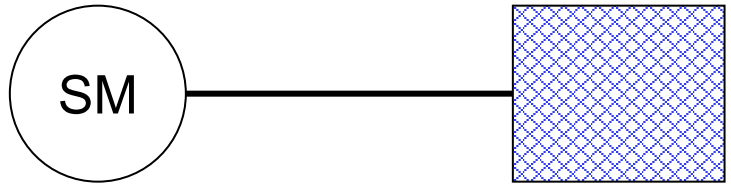
Components:

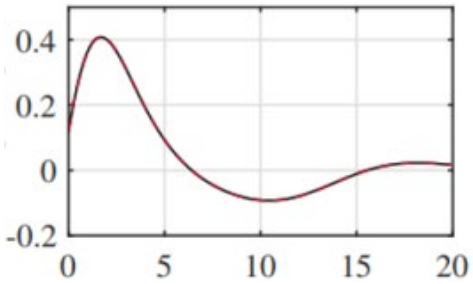
- ODE definition & parameter configuration
- Dataset generation (trajectories + collocation)
- Preprocessing & sampling controls
- PINN training loop (PyTorch, Hydra, Wandb)
- Evaluation & visualization

I. Karampinis, P. Ellinas, I. Ventura-Nadal, R. Nellikkath, S. Chatzivasileiadis, A Toolbox for Physics-Informed Neural Networks in Power Systems, IEEE Powertech 2025, https://arxiv.org/pdf/2502.06412

Results – PINNs trained from the toolbox

Synchronous machine with 9 states





	1 trajectory	50 trajectories	500 trajectories
ODE solver	10.81ms	54.06ms	540.61ms
PINN	1.95ms **(x5.5)**	3.82ms **(x14)**	8.59ms **(x63)**

Key point: PINN scales massively better due to GPU **parallelization**

If we want to create a PINN-based simulator….

Are PINNs scalable?

Can we have a single PINN for 1,000 buses?

Learning takes longer

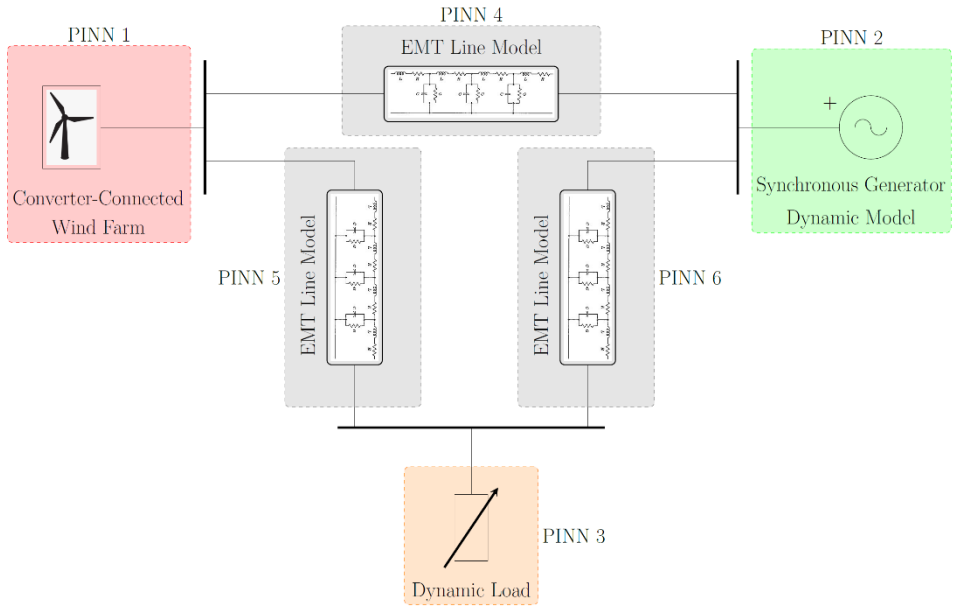
PINN accuracy drops

Short answer: probably not

What shall we do?

Physics-Informed Neural Networks for Power Systems: Vision (Part I)

1. **PINNSim: A modular power system time-domain simulator**
   - A library of component models implemented with Neural Networks
   - "Drag'n'drop" to create your system



- A completely new way of simulation which can be 10x-100x faster.
  - What does this mean? Instead of assessing 100 scenarios leading to a blackout within 1 hour, I can now assess 10,000 scenarios

Very first version of **PINNSim** simulation engine:

J. Stiasny, B. Zhang, S. Chatzivasileiadis, PINNSim: A Simulator for Power System Dynamics based on Physics-Informed Neural Networks, PSCC 2024. https://arxiv.org/abs/2303.10256

Are PINNs Trustworthy?

- They are as trustworthy as any reduced-order model
  - Most reduced-order models come with no guarantees about worst-case violation errors
  - **But,** reduced-order models come from first principles, so we have picked the equations that are relevant to us →we have an intuition which dynamic phenomena we capture and which not

- Work on verifying PINNs
  - If successful, for the first time we will have reduced-order dynamic models
  - Major challenge: how do you verify (= optimize) through differential equations?

DTU

DTU Wind

Physics-Informed Neural Networks for Power Systems: Vision (Part II)

2. Verify PINNs

- For the first time, deliver a worst-case guarantee of the PINN approximation
- Deliver ML Surrogate Models with approximation error guarantees

Efficient Error Certification for Physics-Informed Neural Networks

Francisco Eiras [1] Adel Bibi [1] Rudy Bunel [2] Krishnamurthy Dj Dvijotham [2] Philip H.S. Torr [1]
M. Pawan Kumar [2]

Abstract

Recent work provides promising evidence that Physics-Informed Neural Networks (PINN) can efficiently solve partial differential equations (PDE). However, previous works have failed to provide guarantees on the *worst-case* residual error of a PINN across the spatio-temporal domain – a measure akin to the tolerance of numerical solvers – focusing instead on point-wise comparisons between their solution and the ones obtained by a solver on a set of inputs. In real-world ap-

mentioned challenge through *physics-informed neural networks* (PINN) (Raissi et al., 2019a; Sun et al., 2020; Pang et al., 2019). For example, the Diffusion-Sorption equation – which has real-world applications in the modeling of groundwater contaminant transport – takes 59.83s to solve per inference point using a classical PDE solver, while inference in its PINN version from Takamoto et al. (2022) takes only 2.7×10^{-3}s, a speed-up of more than 10^4 times.

The parameters of a PINN are estimated by minimizing the residual of the given PDE, together with its initial and boundary conditions, over a set of spatio-temporal training

F. Eiras, A. Bibi, R. Bunel, K. Dvijotham, P. Torr, M. P. Kumar, Efficient Error Certification for Physics-Informed Neural Networks, ICML 2024, https://arxiv.org/pdf/2305.10157

Correctness Verification of Neural Networks Approximating Differential Equations

Petros Ellinas [1] Rahul Nellikath [1] Ignasi Ventura [1] Jochen Stiasny [1] Spyros Chatzivasileiadis [1]

Abstract

Verification of Neural Networks (NNs) that approximate the solution of Partial Differential Equations (PDEs) is a major milestone towards enhancing their trustworthiness and accelerating their deployment, especially for safety-critical systems. If successful, such NNs can become integral parts of simulation software tools which can accelerate the simulation of complex dynamic systems more than 100 times. However, the verification of these functions poses major challenges: it is not

providing a formal bound on the lowest accuracy across the relevant input domain. The concept behind correctness guarantees involves determining the worst-case approximation error in the input domain $\mathcal{D}$ and it can be formulated as an optimization problem

$$\max_{x \in \mathcal{D}} |u(x) - u_\theta(x)|, \quad (1)$$

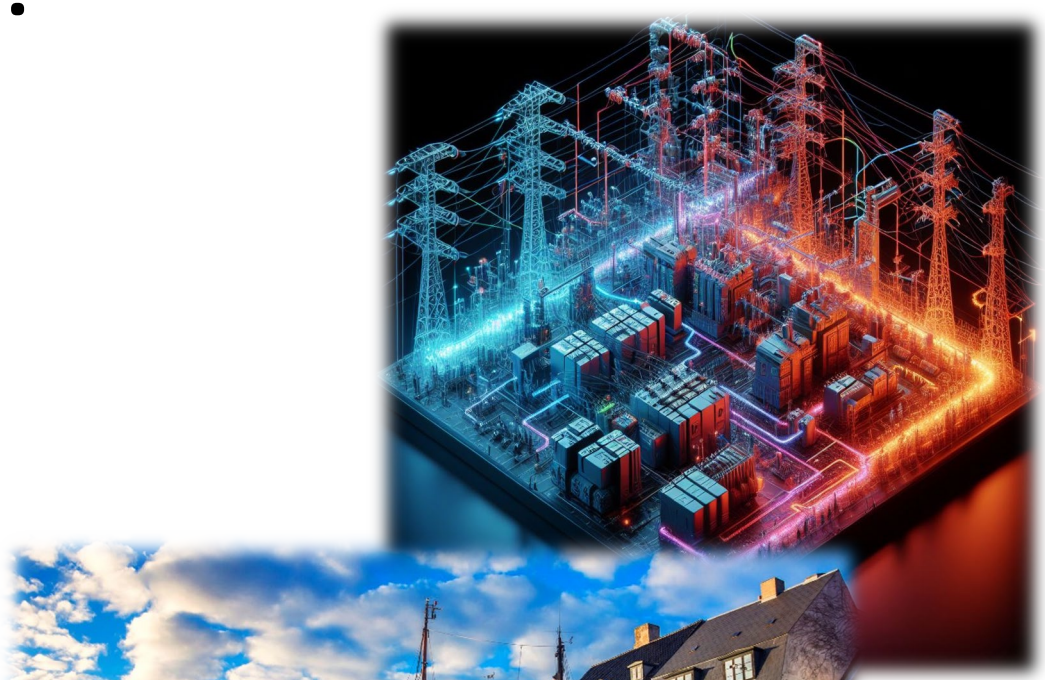
where $u(x)$ is the ground truth solution, and $u_\theta(x)$ is the NN function approximation with weights θ. Here, $x \in \mathcal{D}$ is a point in the input domain $\mathcal{D}$. The argument that

P. Ellinas, R. Nellikkath, J. Stiasny, S. Chatzivasileiadis, Correctness Verification of Neural Networks Approximating Differential Equations, https://arxiv.org/abs/2402.07621

Interested in a PhD or Postdoc on Trustworthy AI and PINNs for Power Systems?

- We have open postdoc positions!

- Send an email to spchatz@dtu.dk



Hands-on Tutorial: Physics Informed Neural Networks for Power Systems

Indrajit Chaudhuri
Intern

Rahul Nellikkath
Postdoc

Ignasi Ventura Nadal
PhD student

Tutorial code: Physics-Informed Neural Networks for Single Machine Infinite Bus

- Google Colab Python Notebook

https://colab.research.google.com/drive/1plxPZf-A4h-sPsFwpmcDbPqClwAwp_g1?usp=sharing#scrollTo=irgglR1yzyIO

Hands-on Tutorial: Physics Informed Neural Networks for Power Systems

1. You need Google credentials. Otherwise, you need to see from the person sitting next to you

2. https://colab.research.google.com/drive/1plxPZf-A4h-sPsFwpmcDbPqClwAwp_g1?usp=sharing#scrollTo=irgglR1yzyIO

3. File → Save a Copy in Drive (so that you can edit the notebook)

4. Run each snippet of code

5. A number of Tasks to perform at the end
   - E.g. Change the PINN Test parameters in "Testing the performance of the neural network"

Questions

1. What is the activation function we use for the neural network?

2. How many layers and how many neurons?

3. How many collocation points?

4. What is the time range of the dynamic phenomenon that we train for?

5. How many inputs? How many outputs?
   - Which are those?

6. How many epochs do we train for?

7. How many initial conditions do we train for?
   - What is their range?

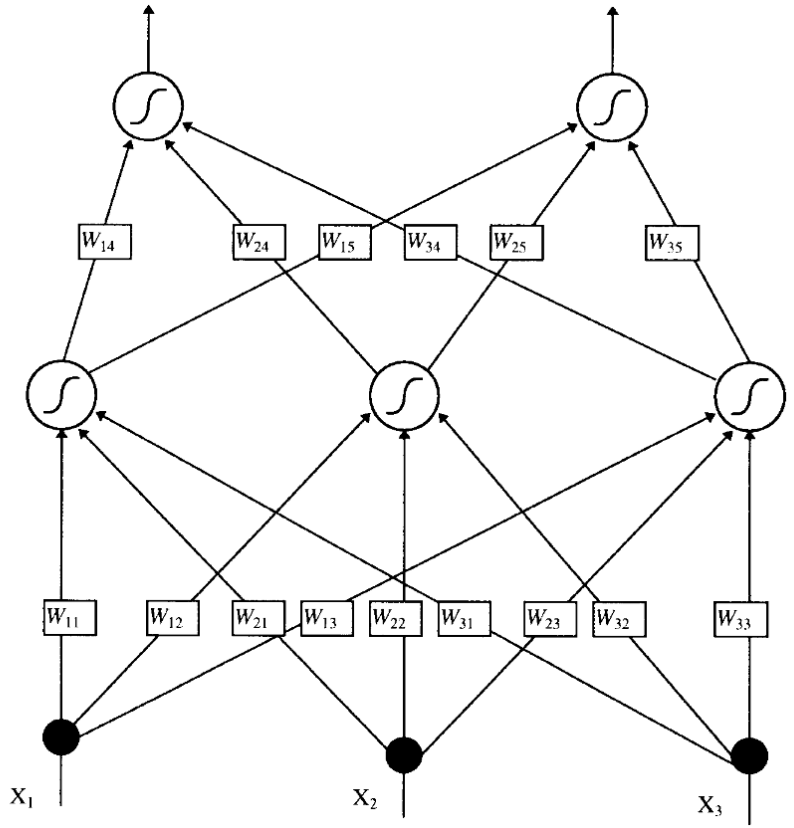
Trustworthy AI for Power Systems

AI is already creating value in Energy Systems

- Load Forecasting
- Weather Forecasting
- Predictive Maintenance
- Energy Trading (forecasting of prices or quantities)

AI is already creating value in Energy Systems

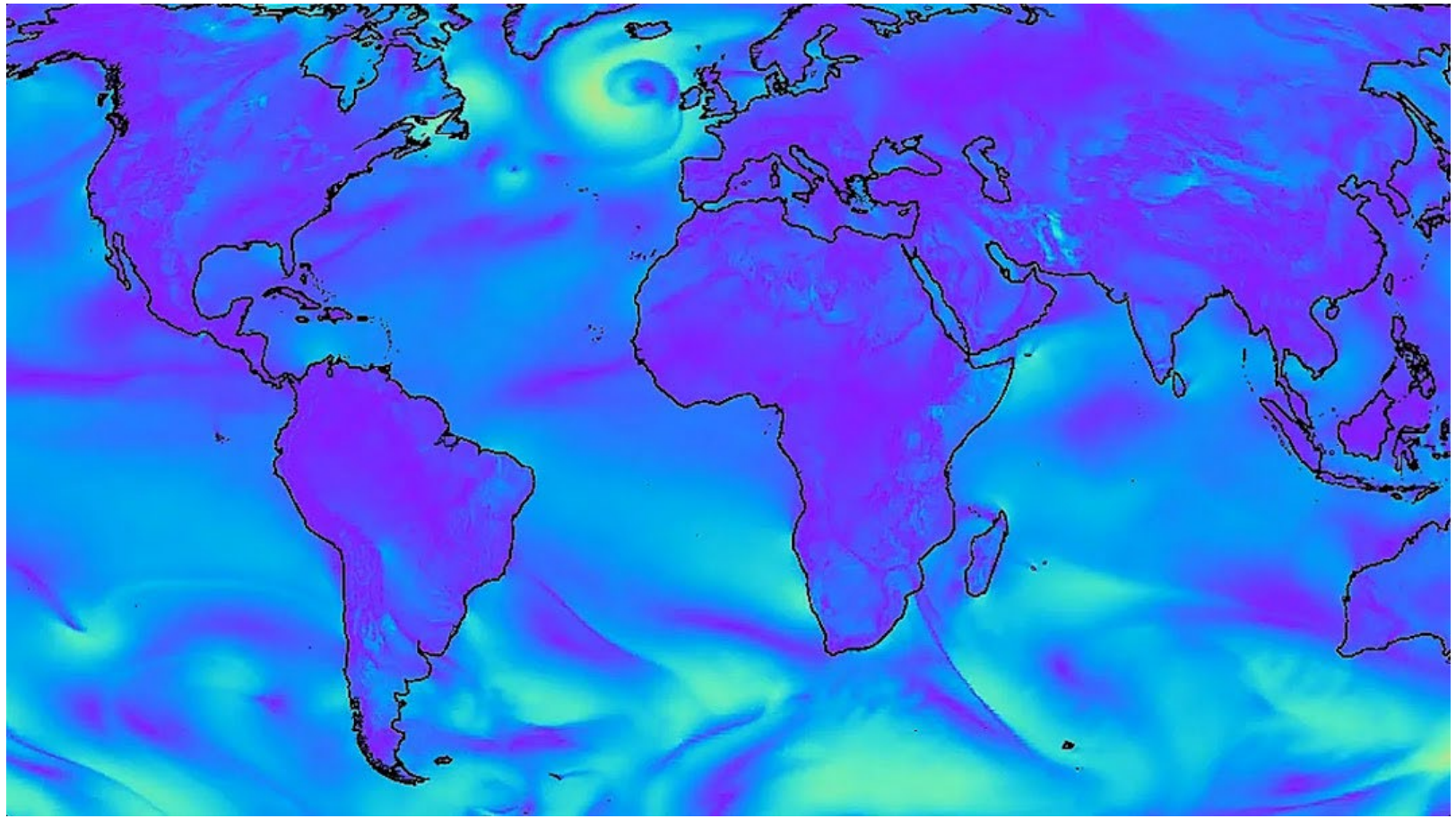
- **Load Forecasting**
- Weather Forecasting
- Predictive Maintenance
- Energy Trading (forecasting of prices or quantities)



- **ANNSTLF**: Probably the **first tool based on Machine Learning** in Power Systems
- Developed by EPRI (Electric Power Research Institute) in the US
- First deployed in 1992 in Texas. Deployed to 32 utilities by 1997

AI is already creating value in Energy Systems

- Load Forecasting
- **Weather Forecasting**
- Predictive Maintenance
- Energy Trading (forecasting of prices or quantities)



Google Graphcast: AI is already better than physical models for global weather forecasting

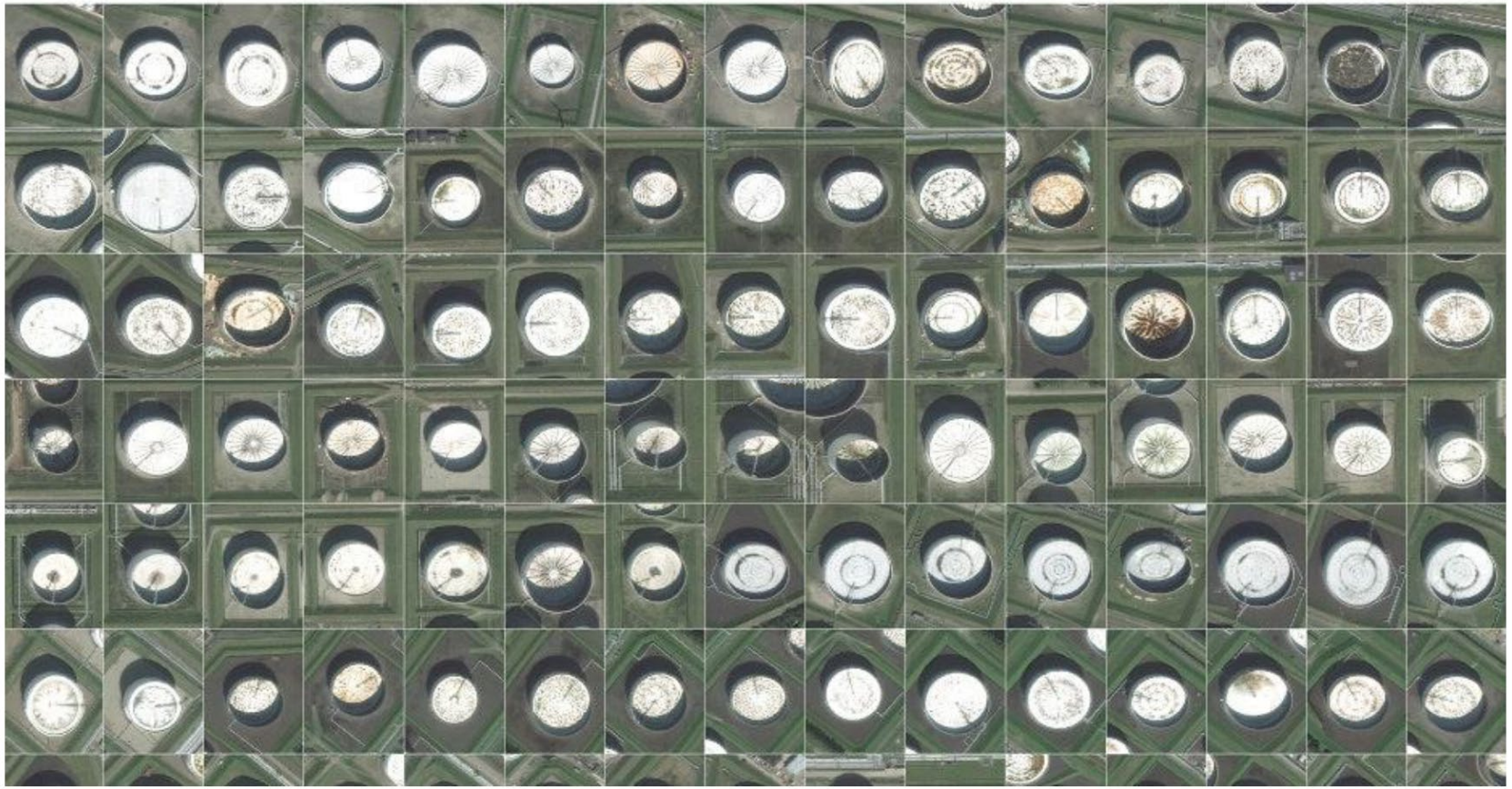
AI is already creating value in Energy Systems

- Load Forecasting
- Weather Forecasting
- **Predictive Maintenance**
- Energy Trading (forecasting of prices or quantities)

- Combination of images with other sensor data to predict failures
- IEA: digitalization can help lower maintenance costs of electricity grids by 5% = **80 billion EUR**

AI is already creating value in Energy Systems

- Load Forecasting
- Weather Forecasting
- Predictive Maintenance
- **Energy Trading** (forecasting of prices or quantities)



But AI can do a lot more things

1. Process massive amounts of texts (e.g. regulations, manuals, procedures, etc)
2. Virtual assistant: Helping maintenance technicians with step-by-step instructions
3. Support for decision making

And many more

But: Would you ever trust AI to run your electricity network?

But: Would you ever trust AI to run your electricity network?

What would you do to make it trustworthy?

Making AI Trustworthy: My View

Verify AI

Making it safe to deploy as is

Performance Guarantees

1. your AI tool will never violate the voltage constraints
2. Or, your AI tool will violate the voltage constraint by XX % in the worst-case

European Research Council
Established by the European Commission

veriphied.ai

Horizon Europe

ai-effect.eu/

Making AI Trustworthy: My View

Verify AI

Making it safe to deploy as is

Performance Guarantees

1. your AI tool will never violate the voltage constraints
2. Or, your AI tool will violate the voltage constraint by XX % in the worst-case

veriphied.ai

ai-effect.eu/

Use AI as Decision Support

1

Take the AI output and project it to a **feasible space**

2

Use the AI output as a **warm-start** for an optimizer (or to predict the active constraints)

3

Use the AI to **screen** millions of scenarios. Assess with conventional tools the most critical ones

Making AI Trustworthy: My View

Verify AI

Making it safe to deploy as is

Performance Guarantees

1. your AI tool will never violate the voltage constraints
2. Or, your AI tool will violate the voltage constraint by XX % in the worst-case

erc
European Research Council
Established by the European Commission

veriphied.ai

Horizon Europe

ai-effect.eu/

Use AI as Decision Support

1

Take the AI output and project it to a **feasible space**

2

Use the AI output as a **warm-start** for an optimizer (or to predict the active constraints)

3

Use the AI to **screen** millions of scenarios. Assess with conventional tools the most critical ones

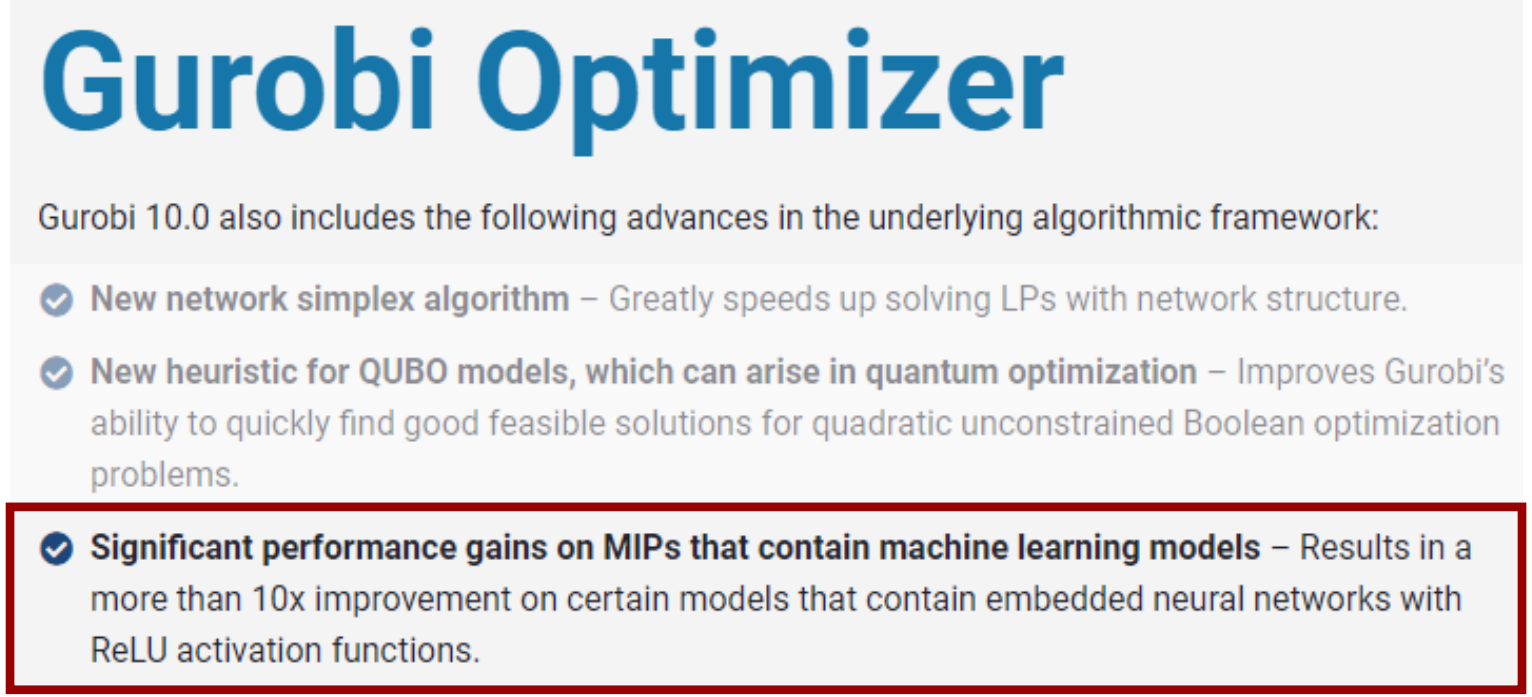
A lot of recent developments for trustworthy AI

- **April 2021:** The EU is promoting rules for Trustworthy AI

- Visit of Ms. Margrethe Vestager at DTU
  - EU Commissioner of Competition, Executive Vice President of "A Europe Fit for the Digital Age"
  - In April 2021, Ms. Vestager proposed new rules and actions aiming to turn Europe into the global hub for trustworthy Artificial Intelligence

A lot of recent developments for trustworthy AI

- World-leading optimization tool: Starting with Gurobi 10.0, Gurobi supports Neural Network verification since 2023



A lot of recent developments for trustworthy AI

6th International Verification of Neural Networks Competition (VNN-COMP'25)

https://sites.google.com/view/vnn2025

- Tailored MILP solvers for NN Verification
  - Alpha-beta-crown is the winning algorithm
  - Over 100x speedup
- Focus is mostly on Image Classification/ Image Recognition
  - Key for medical applications such as recognition of MRI images, for self-driving car applications, and others
- **There is an effort to submit models related to power systems, so that participants can test and develop verification algorithms with focus on power systems** (we also tried to submit some power system models, but we did not manage to complete our effort)

Accelerating Verification: α,β-CROWN for DC-OPF

Table 1. Performance comparison of Gurobi and α, β-CROWN solvers on the IEEE 300-bus test case.

Gen scale	Gurobi		α, β-CROWN
	Gap, %	Time, sec	Time, sec
0.8 ✗	26.9	43	6.00
0.9 ✗	79.0	37	5.60
1.0 ✗	150	35	5.71
1.1 ✗	511	39	5.58
1.2 ✓	1726	>3600 (**dnf**)	18.03

α,β-CROWN 7x-300x faster than Gurobi

- We formulated the power system verification problem in a way that can be solved by α,β-CROWN.
- α,β-CROWN now verifies for multiple line flow violations and not only one at a time
- α,β-CROWN much faster than Gurobi 10.0

S. Chevalier, I. Murzakhanov, S. Chatzivasileiadis, *GPU-Accelerated Verification of Machine Learning Models for Power Systems,* **Best Paper Award at HICSS** (Hawaii International Conferences on Systems Sciences), Jan. 2024 https://arxiv.org/pdf/2306.10617

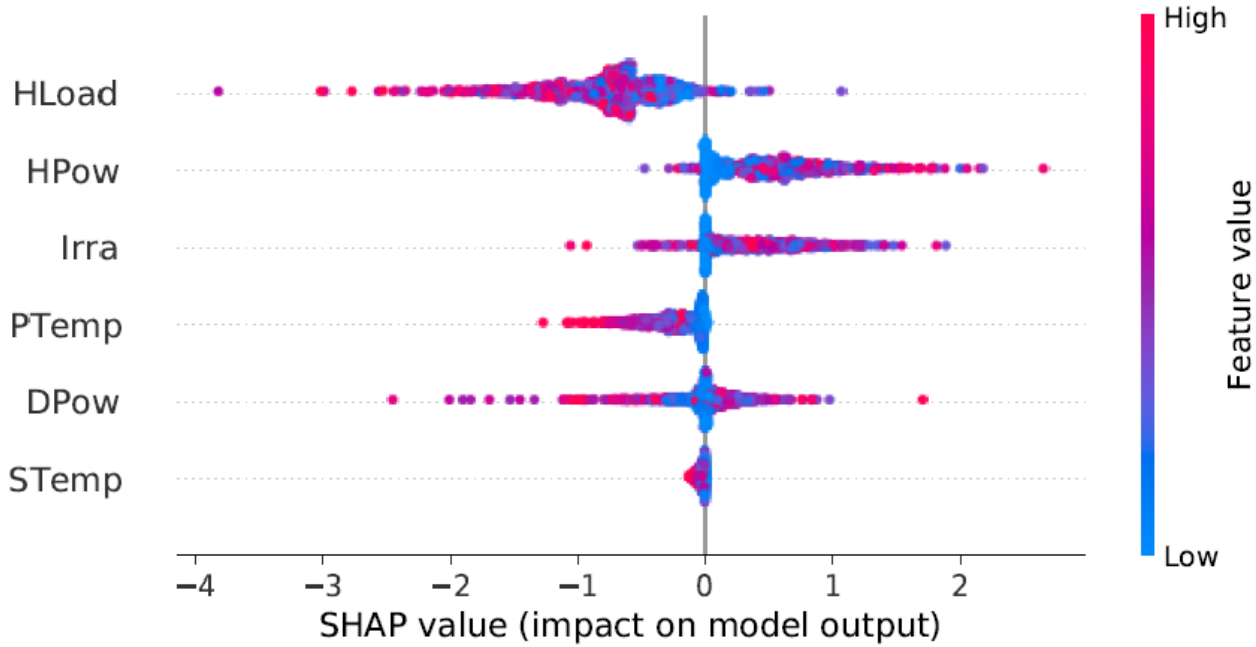
A lot of recent developments for trustworthy AI

- Interpretable AI
- SHAP: Shapley Additive Explanations
- Sensitivity Factors that explain the output of a model

SHAP

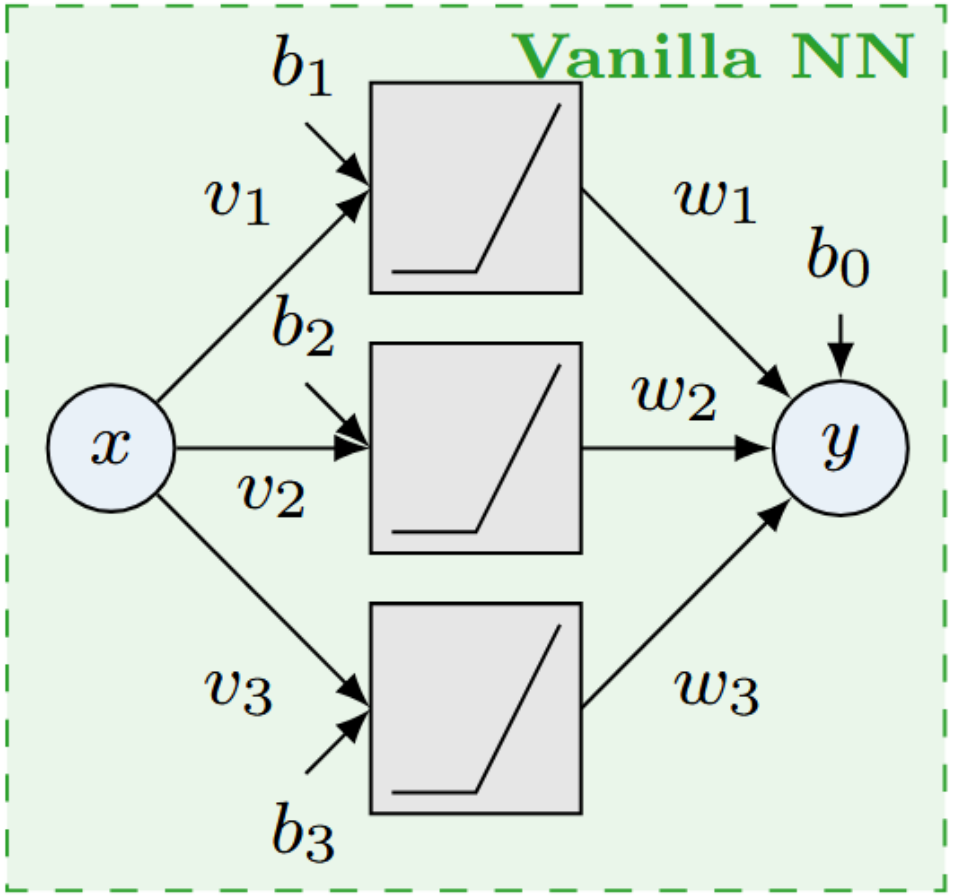
https://shap.readthedocs.io/en/latest/

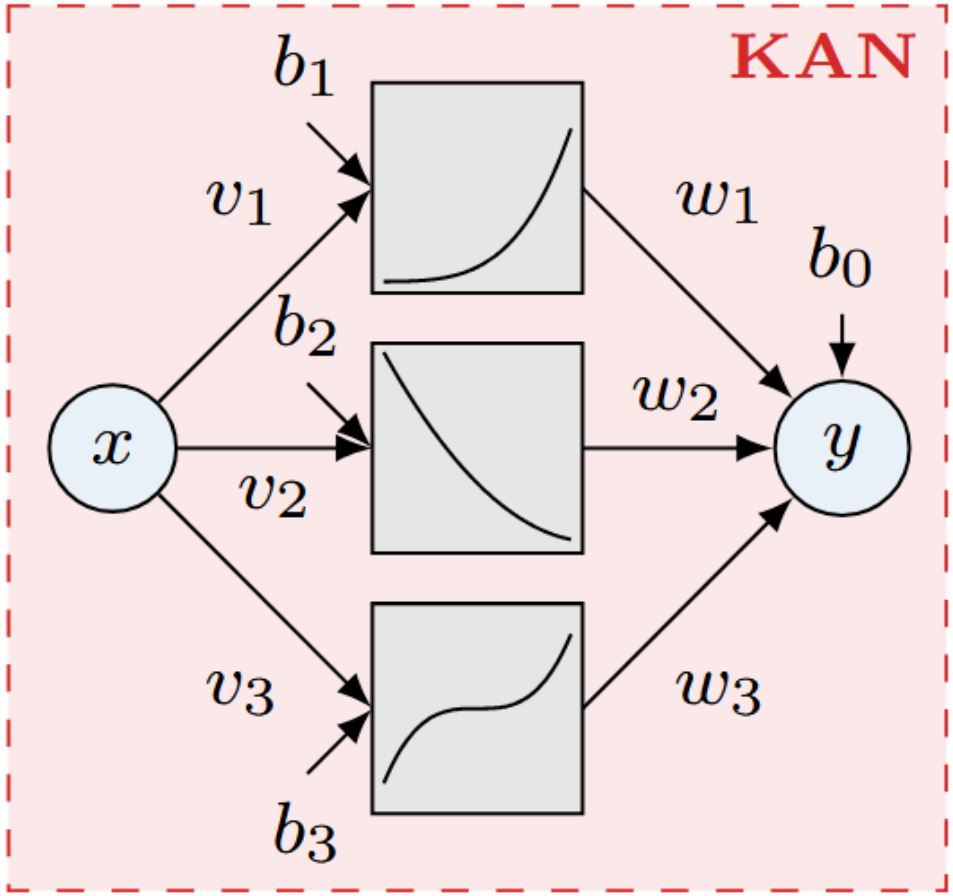
- Predicting the net production of PV+Load
- Positive → PV>load; Negative PV<load



Y. Lu, I. Murzakhanov, S. Chatzivasileiadis, *Neural network interpretability for forecasting of aggregated renewable generation*. In *IEEE SmartGridComm 2021*, Aachen, Germany, October 2021. [.pdf | code]

Kolmogorov Arnold Networks

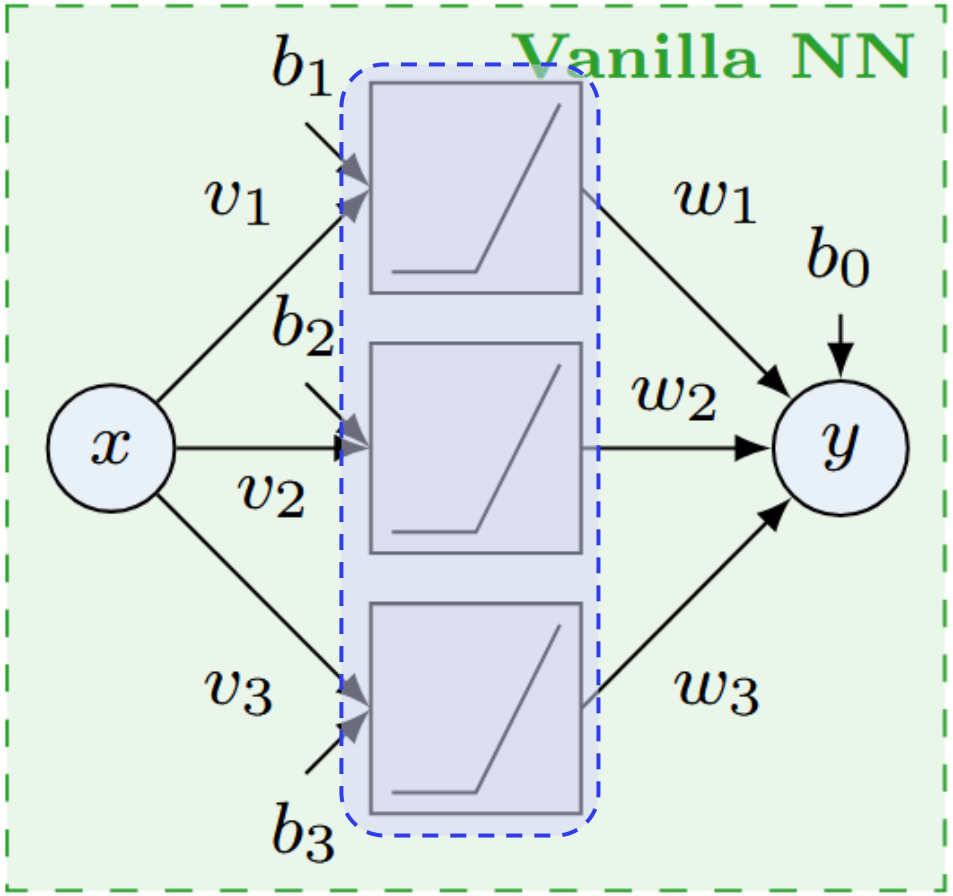


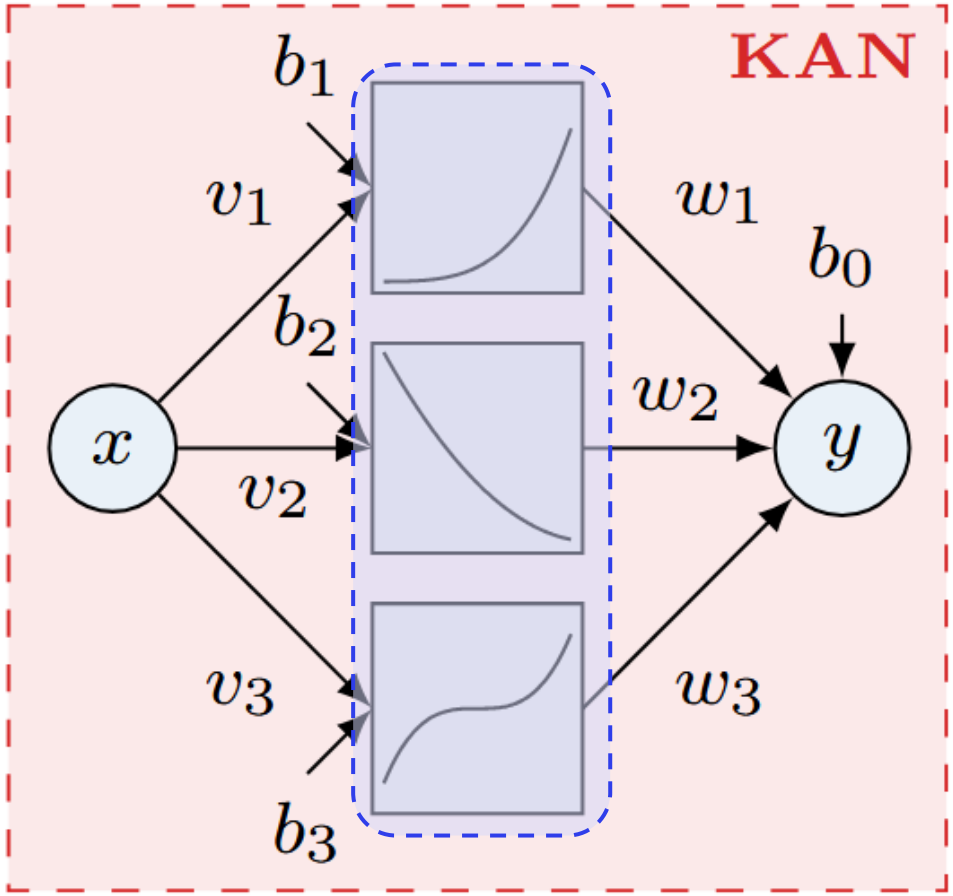


H. Shuai, F. Li, Physics-Informed Kolmogorov-Arnold Networks for Power System Dynamics, https://arxiv.org/pdf/2408.06650

P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. https://doi.org/10.1016/j.segan.2025.101818

Kolmogorov Arnold Networks





H. Shuai, F. Li, Physics-Informed Kolmogorov-Arnold Networks for Power System Dynamics, https://arxiv.org/pdf/2408.06650

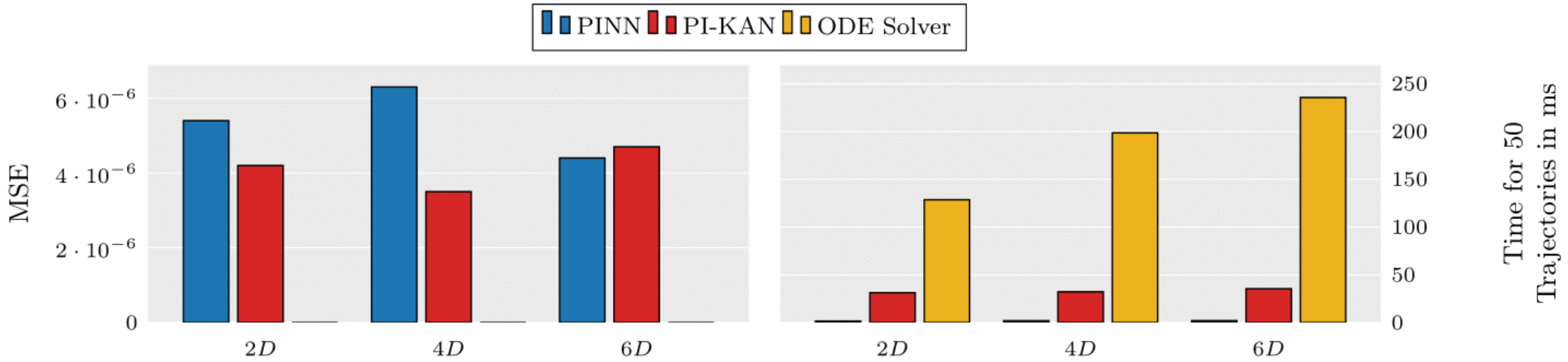
P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. https://doi.org/10.1016/j.segan.2025.101818

Kolmogorov Arnold Networks

- KANs are potentially more interpretable than PINNs (less neurons, trained activation functions which can give some insights)

In our tests:

- KANs are **more accurate** than PINNs
- KANs are **slower** than PINNs



P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. https://doi.org/10.1016/j.segan.2025.101818

Preview:
What are the main takeaways of this final part?

1. **Energy systems are safety-critical systems.** When you develop AI (or OR) approaches think: **can my method be trusted?** What shall I do to guarantee a safe operation?

2. AI needs OR.
   - Major challenges: tractability for realistic size power system problems. There are a lot of cool tricks we can invent to scale these methods

3. **Do not try to reinvent the wheel.** Why should you train an RL agent assuming no prior knowledge, when you e.g. have a detailed model of the battery you want to control? **Develop methods that combine the strengths of physics-based and data-driven methods**.

Neural Network Verification
for Power Systems

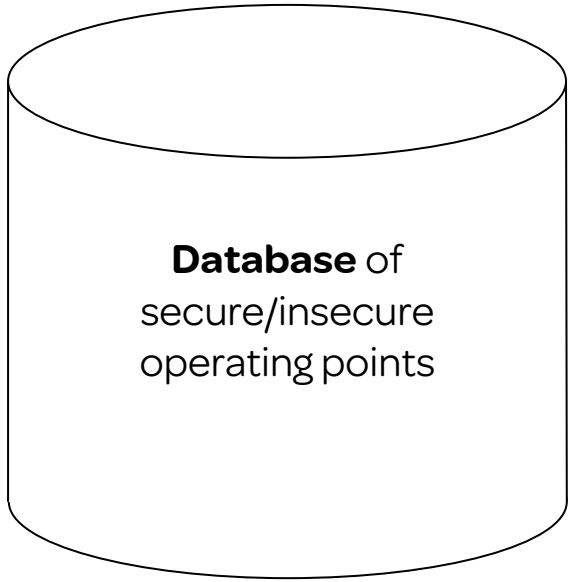
A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. In *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383-397, Jan. 2021, https://arxiv.org/pdf/1910.01624.pdf

V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in International Conference on Learning Representations (ICLR 2019), 2019

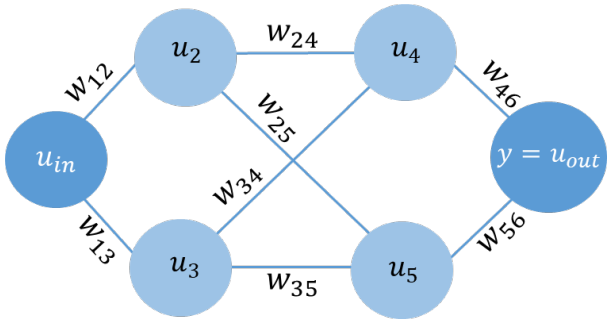
Guiding Application: Security Assessment with Neural Networks

European Research Council

Approaches proposed up to now



1. Split the database in a training set and a test set

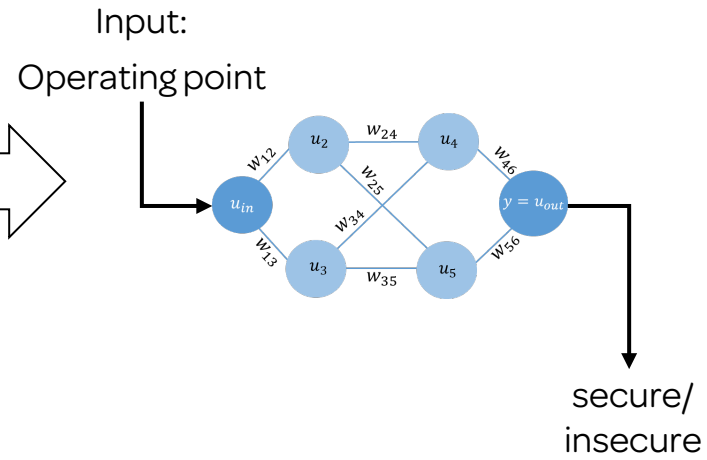


2. Train a neural network

3. Test the neural network

4. Is accuracy high enough?

5. Use the NN



NN Output:

Binary classification: **secure/insecure**

Extremely fast: up to 100x-1'000x faster

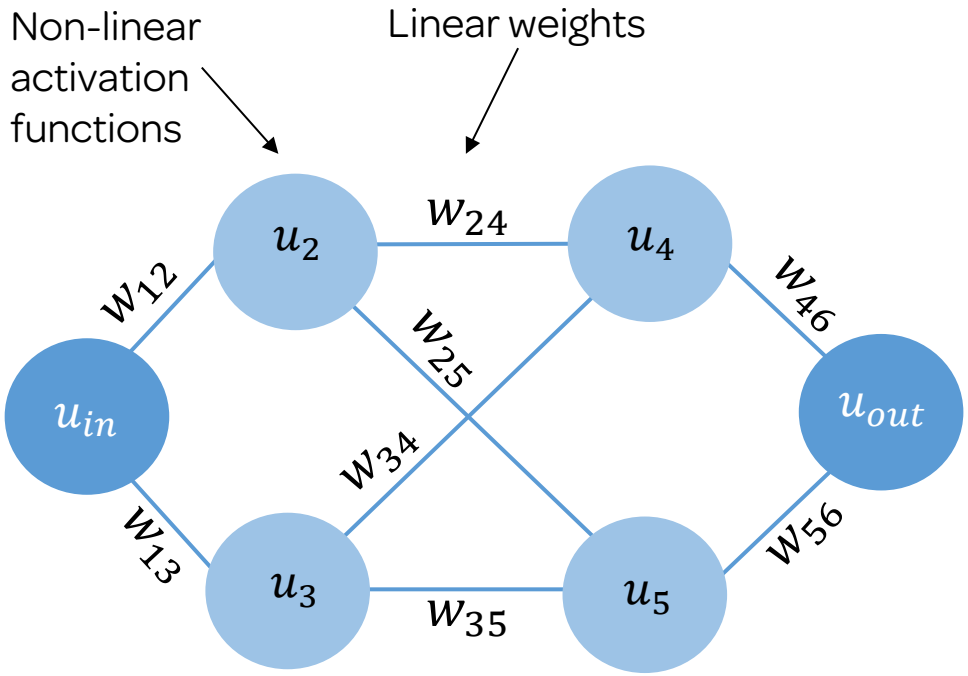
Neural Network Verification: HOW?

1. **Exact transformation:** Convert the neural network to a **set of linear equations with binary variables**
   - The Neural Network can be included in a mixed-integer linear optimization problem

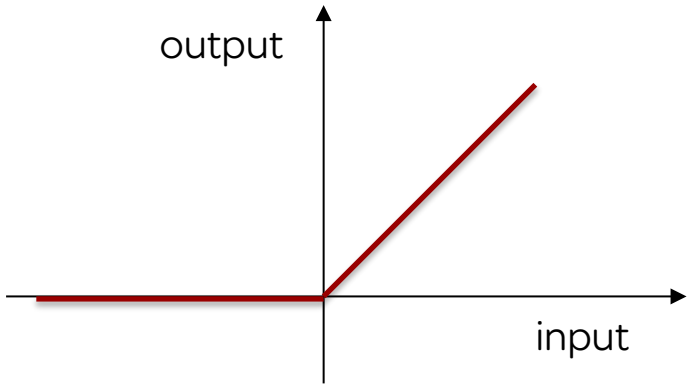
2. Formulate an **optimization** problem and solve it → certificate for NN behavior

3. Assess if the neural network output complies with the ground truth

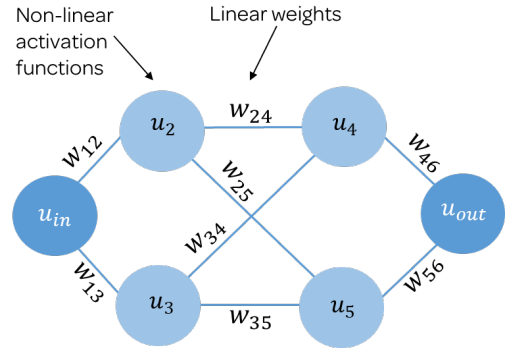
From Neural Networks to Mixed-Integer Linear Programming



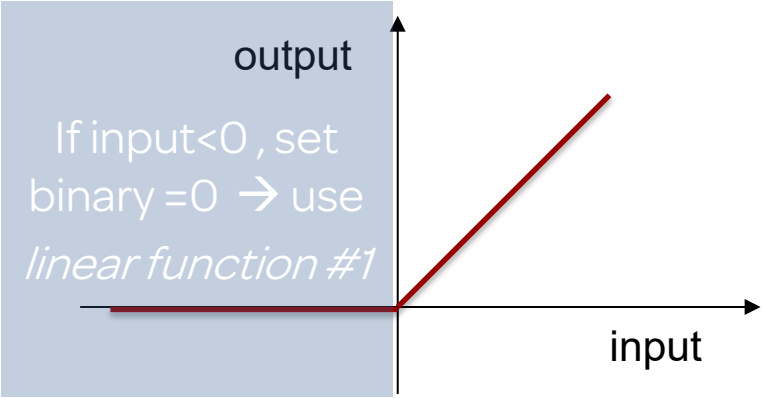
- Most usual activation function: ReLU
- **ReLU:** Rectifier Linear Unit



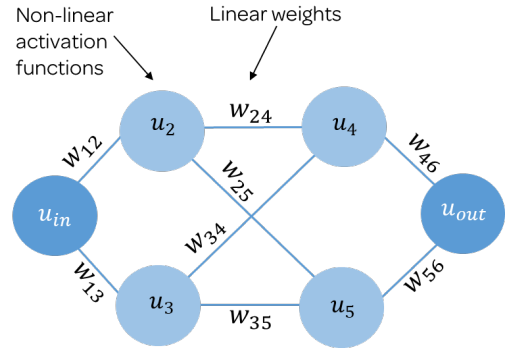
From Neural Networks to Mixed-Integer Linear Programming



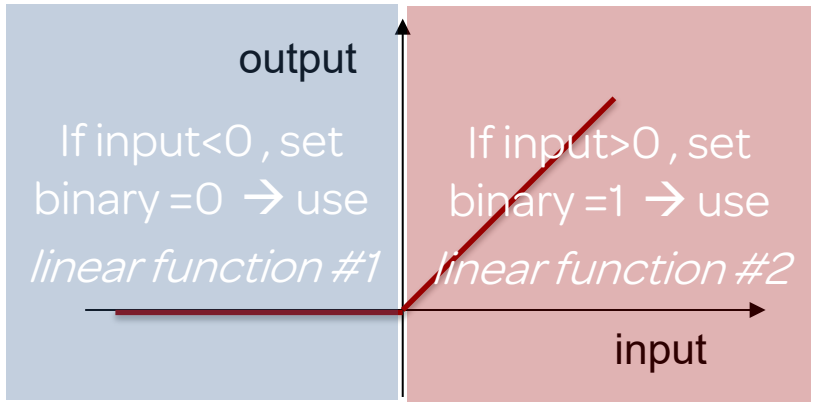
1. But **ReLU** can be transformed to a **piecewise linear function with binary variables**



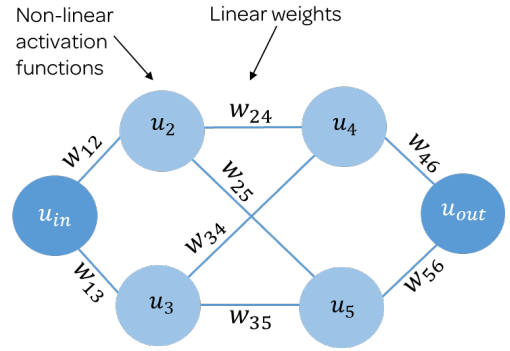
From Neural Networks to Mixed-Integer Linear Programming



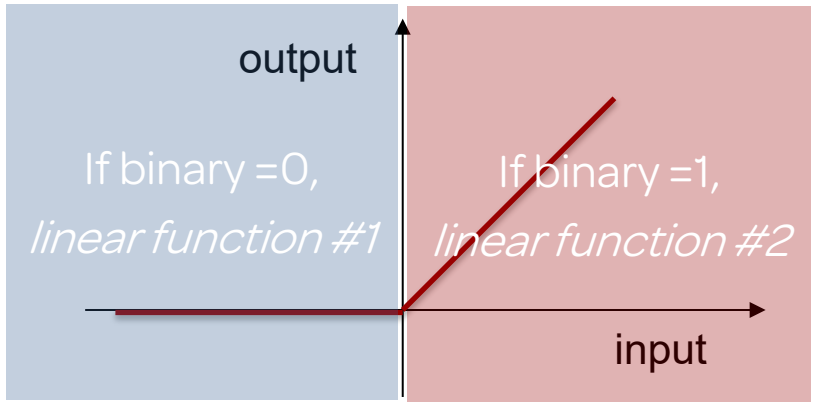
1. But **ReLU** can be transformed to a **piecewise linear function with binary variables**



From Neural Networks to Mixed-Integer Linear Programming



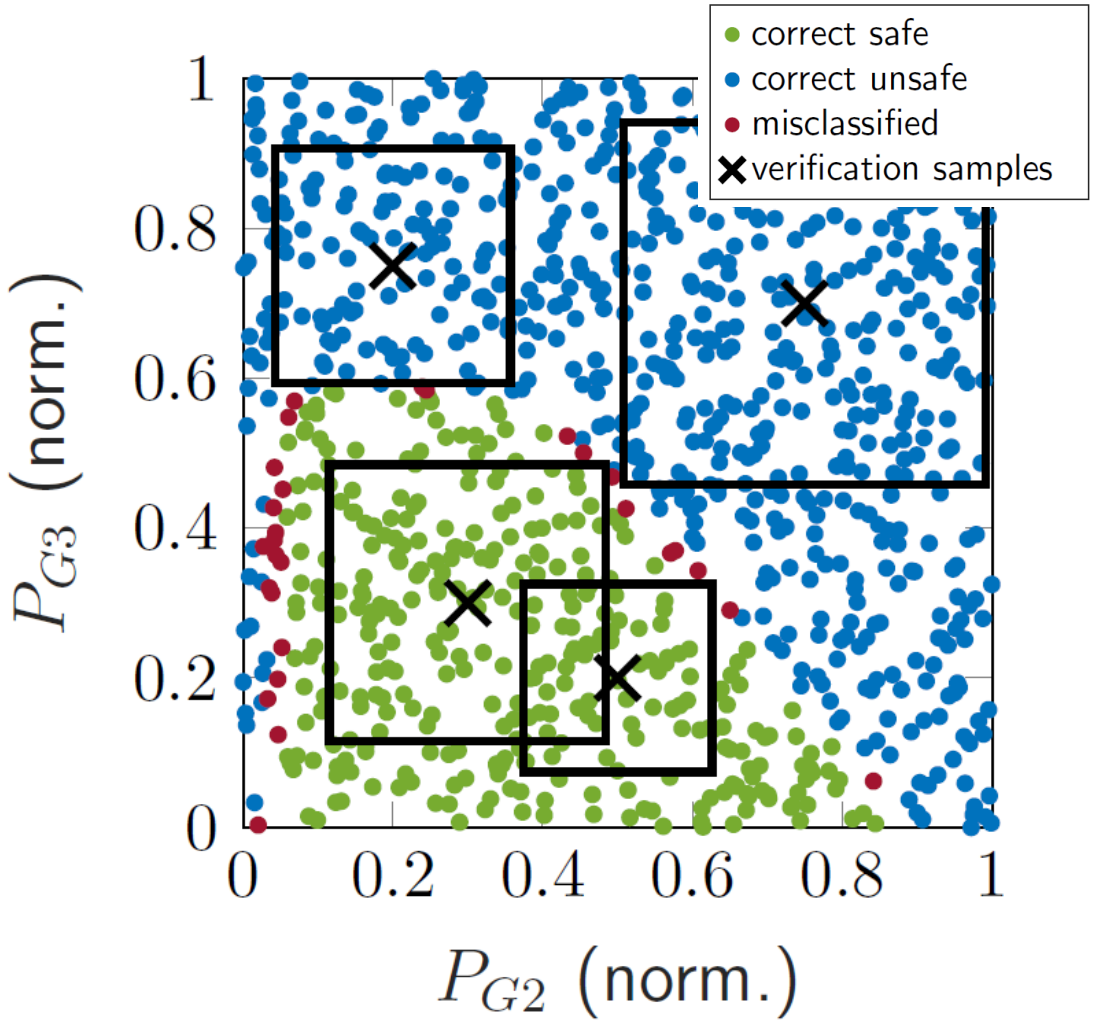
1. But **ReLU** can be transformed to a **piecewise linear function with binary variables**



2. I can encode all operations of a Neural Network to a system of linear equations with continuous and binary variables

3. I can **integrate** all information encoded in a **neural network inside an optimization program**

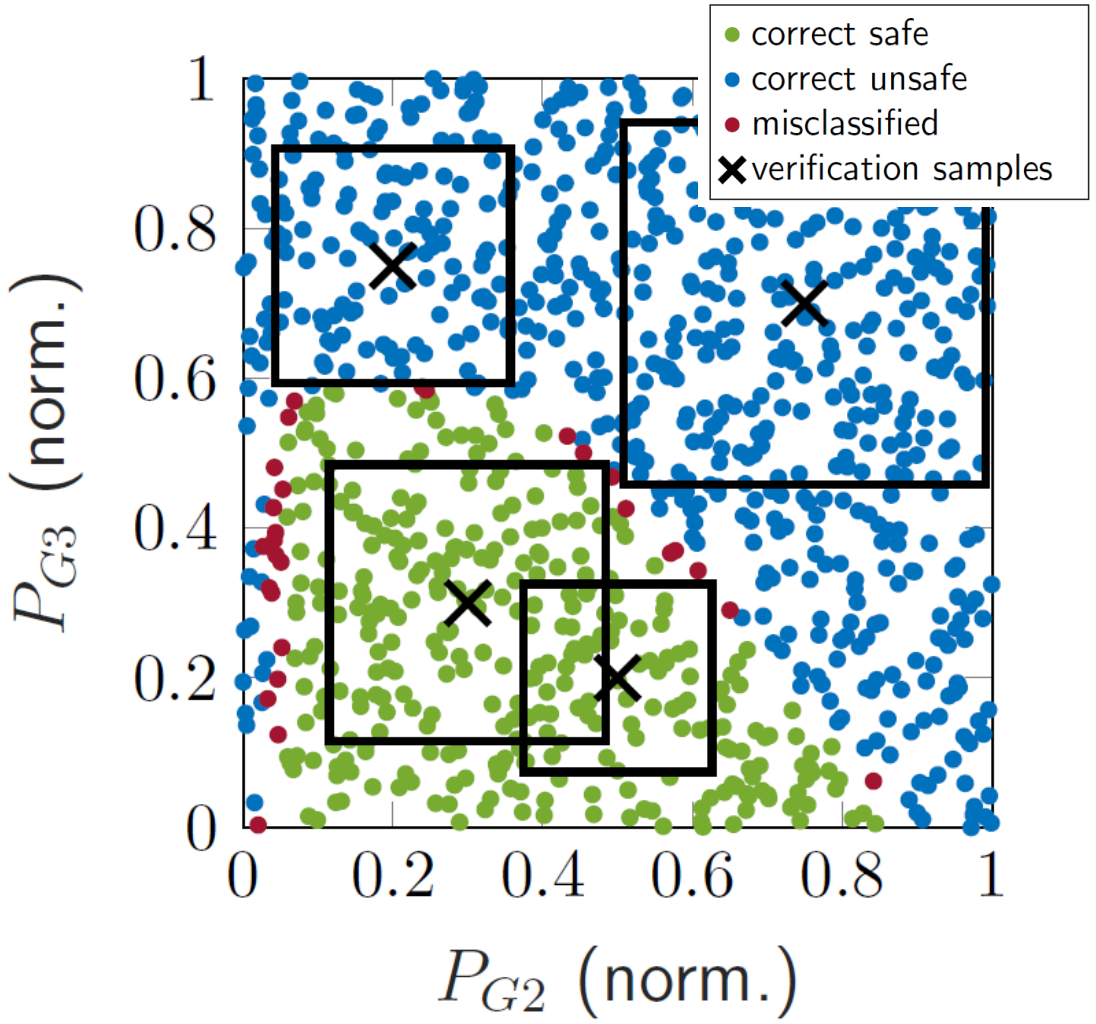
Certify the output for a continuous range of inputs



1. We assume a given input x_{ref} with classification "safe"

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *IEEE Transactions on Smart Grid*, Jan. 2021. https://arxiv.org/pdf/1910.01624.pdf

Certify the output for a continuous range of inputs



1. We assume a given input x_{ref} with classification "safe"
2. Solve optimization problem: **Does classification change for *any* input within distance ε from x_{ref}?**
3. If not, then **I can certify** that my neural network will classify the whole continuous region as "safe"
4. I can repeat this for other regions and different classifications

A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *IEEE Transactions on Smart Grid*, Jan. 2021. https://arxiv.org/pdf/1910.01624.pdf

Provable Worst-case Guarantees

Venzke, G. Qu, S. Low, S. Chatzivasileiadis, Learning Optimal Power Flow: Worst-case Guarantees for Neural Networks. **Best Student Paper Award** at IEEE SmartGridComm 2020. https://arxiv.org/pdf/2006.11029.pdf

R. Nellikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks for Minimising Worst-Case Violations in DC Optimal Power Flow. In IEEE SmartGridComm 2021, Aachen, Germany, October 2021.

R. Nellikkath, S. Chatzivasileiadis. Physics-Informed Neural Networks for AC Optimal Power Flow. 2021.

Key Enabler: our ability to represent the underlying ground truth

Main idea:

- Take advantage of the ground truth representation we have, i.e. the power system models
- Measure the performance of the Neural Network against the ground truth
  - Does the Neural Network violate constraints?
- **Determine the worst-case performance = provable worst-case guarantees**
  - **Across the continuous input domain**
  - No Sampling
  - Instead, we solve an optimization program
  - Once "certified", we can use directly the Neural Network (no need to re-run the optimization program)

Worst violation over the **whole training dataset** (training+test set)

Our algorithm: **provable** worst-case guarantee over the **whole input domain**

	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_{g} (MW)	ν_{line} (MW)	ν_{g} (MW)	ν_{line} (MW)
case9				
case30				
case39				
case57				
case118				
case162				
case300				

ν_{g} Maximum violation of generator limits

ν_{line} Maximum violation of line limits

Worst violation over the **whole training dataset** (training+test set)

Our algorithm: **provable** worst-case guarantee over the **whole input domain**

	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_{g} (MW)	ν_{line} (MW)	ν_{g} (MW)	ν_{line} (MW)
case9	2.5	1.8	2.8	1.9
case30	1.7	0.6	3.6	3.1
case39	51.9	37.2	270.6	120.0
case57	4.2	0.0	23.7	0.0
case118	149.4	15.6	997.8	510.8
case162	228.0	180.0	1563.3	974.1
case300	474.5	692.7	3658.5	3449.3

ν_{g} Maximum violation of generator limits

ν_{line} Maximum violation of line limits

Over the whole input domain **violations can be much larger** (here ~7x) compared to what has been estimated empirically on the dataset

Worst violation over the **whole training dataset** (training+test set)

New algorithm: **provable** worst-case guarantee over the **whole input domain**

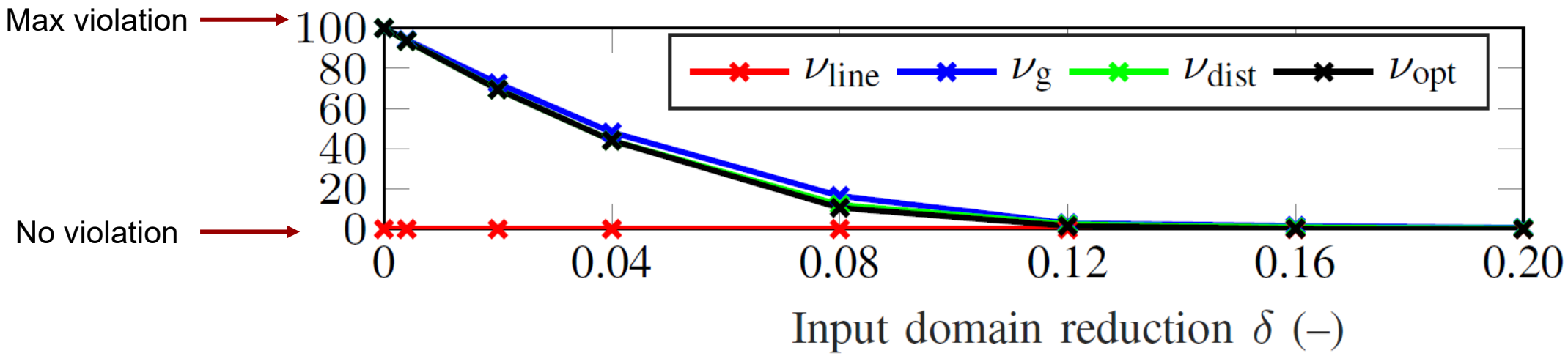
	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_g (MW)	ν_{line} (MW)	ν_g (MW)	ν_{line} (MW)
case9	2.5	1.8	2.8	1.9
case30	1.7	0.6	3.6	3.1
case39	51.9	37.2	270.6	120.0
case57	4.2	0.0	23.7	0.0
case118	149.4	15.6	997.8	510.8
case162	228.0	180.0	1563.3	974.1
case300	474.5	692.7	3658.5	3449.3

ν_g Maximum violation of generator limits

ν_{line} Maximum violation of line limits

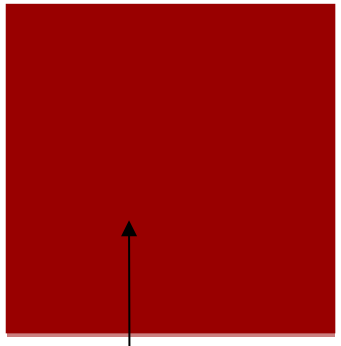
We can now provide **guarantees that no NN output will violate the line limits** over the whole input domain

Reducing the worst-case violations

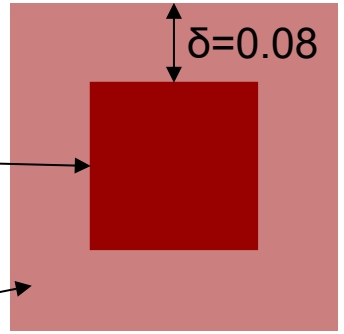


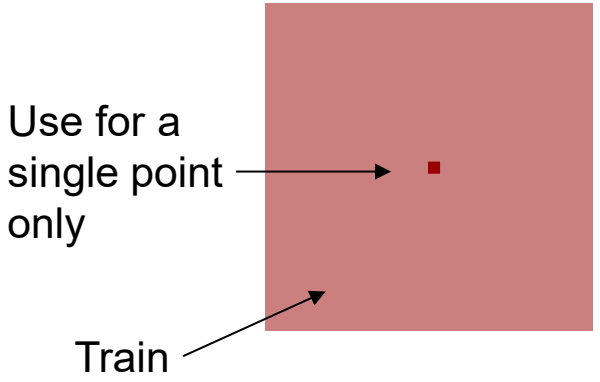
Training Dataset:

Load Domain [60%-100%]



Train and use on the same domain





What shall we do next?

1. Integrate worst-case violations in NN training?

2. Graph-Neural Networks for N-k Security Assessment?

3. Play with a PINN?
   - Google Colab Notebook

4. Conclude and Discuss?

What shall we do next?

1. Integrate worst-case violations in NN training?
2. Graph-Neural Networks for N-k Security Assessment?
3. Play with a PINN?
   - Google Colab Notebook
4. Conclude and Discuss?

Integrating Worst-Case Violations in NN training -- Begin

Worst-case Violations: What is the next natural step?

Integrate the worst-case violations *inside* the neural network training procedure

Our "Holy Grail": Design a **Neural Network training procedure** that:

- produces a Neural Network with best average performance,
- *and* delivers guarantees about its worst-case performance

Worst-case Violations: What is the next natural step?

Integrate the worst-case violations *inside* the neural network training procedure

Our "Holy Grail": Design a **Neural Network training procedure** that:

- produces a Neural Network with best average performance,
- ***and*** delivers guarantees about its worst-case performance

(Random) Example of an imaginary final message:

- "Neural Network Training finished. Accuracy 99.2%. Worst-case violation of critical constraints: 10%."

Wouldn't that create a good level of trust for applying NNs on any safety-critical system?

Extends beyond power systems → drones, air-traffic control, robots, control of inverters, and others

How can we integrate worst-case violations in NN training?

- Standard NN training

$$\min_{\mathbf{w},\mathbf{b}} \mathcal{L}_0 \equiv \min_{\mathbf{w},\mathbf{b}} \frac{1}{N} \sum_i \| x_i - \hat{x}_i \|$$

How can we integrate worst-case violations in NN training?

- Standard NN training

$$\min_{\mathbf{w},\mathbf{b}} \mathcal{L}_0 \equiv \min_{\mathbf{w},\mathbf{b}} \frac{1}{N} \sum_i \| x_i - \hat{x}_i \|$$

- NN training which penalizes constraint violations
  - Reduces the violations for the training dataset

$$\min_{\mathbf{w},\mathbf{b}} \quad \Lambda_0 \mathcal{L}_0 + \Lambda_p \mathcal{L}_p$$

e.g. $\mathcal{L}_p = v_g = (p_g - p_g^{max})$ *for generator constraint violations*

See Fioretto, Mak, Van Hentenryck, AAAI, 2020, and others

How can we integrate worst-case violations in NN training?

- Standard NN training

$$\min_{\mathbf{w},\mathbf{b}} \mathcal{L}_0 \equiv \min_{\mathbf{w},\mathbf{b}} \frac{1}{N} \sum_i \| x_i - \hat{x}_i \|$$

- NN training which penalizes constraint violations
  - Reduces the violations for the training dataset

See Fioretto, Mak, Van Hentenryck, AAAI, 2020, and others

$$\min_{\mathbf{w},\mathbf{b}} \quad \Lambda_0 \mathcal{L}_0 + \Lambda_p \mathcal{L}_p$$

e.g. $\mathcal{L}_p = v_g = (p_g - p_g^{max})$ *for generator constraint violations*

- **NN training which penalizes worst-case violations**
  - **Worst-case violations might be on datapoints that do not belong to the training dataset.** And we might just discover it when we deploy the NN in a real application
    - this is a major fear of any power system operator (and a main barrier for the NNs in safety-critical applications)

$$\min_{\mathbf{w},\mathbf{b}} \quad \Lambda_0 \mathcal{L}_0 + \Lambda_w \mathcal{L}_{wc}$$

$$\mathcal{L}_{wc} = \max_{\mathbf{D}} \quad v_g$$

Hard bilevel optimization problem

1. Lower level is a MILP
2. **The MILP must be differentiable** so that the NN training can backpropagate

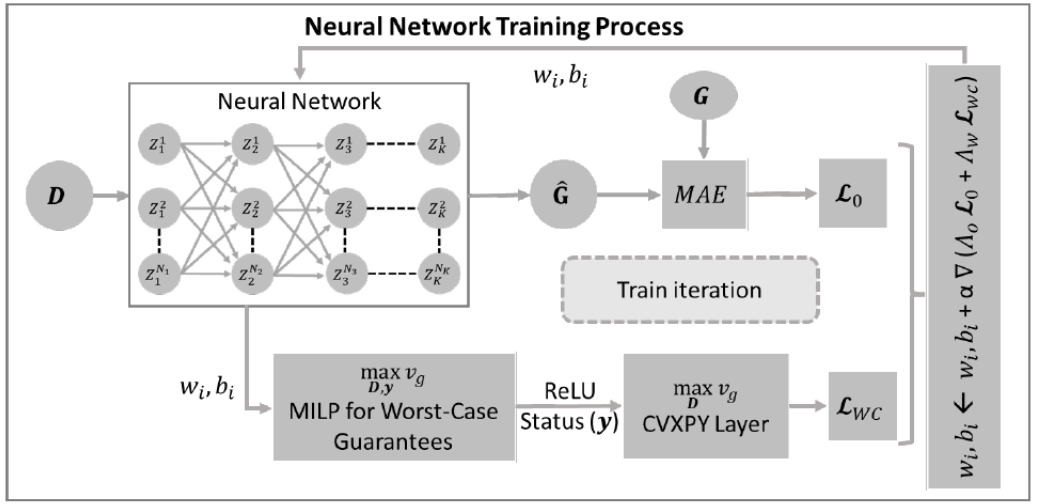
Some thoughts

on how to design an NN training that minimizes worst-case violations

1. Fix the binaries
   - Arbitrary assumption (but it works): for small perturbations of weights & biases, binaries remain constant
   - Solve the lower level MILP by itself, find the binary values for the max constraint violation and fix them

2. MILP is converted to an LP → it is now differentiable

3. Cast it as a differentiable optimization layer (we use CVXPY) → NN training can now backpropagate through it



R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf

Some thoughts

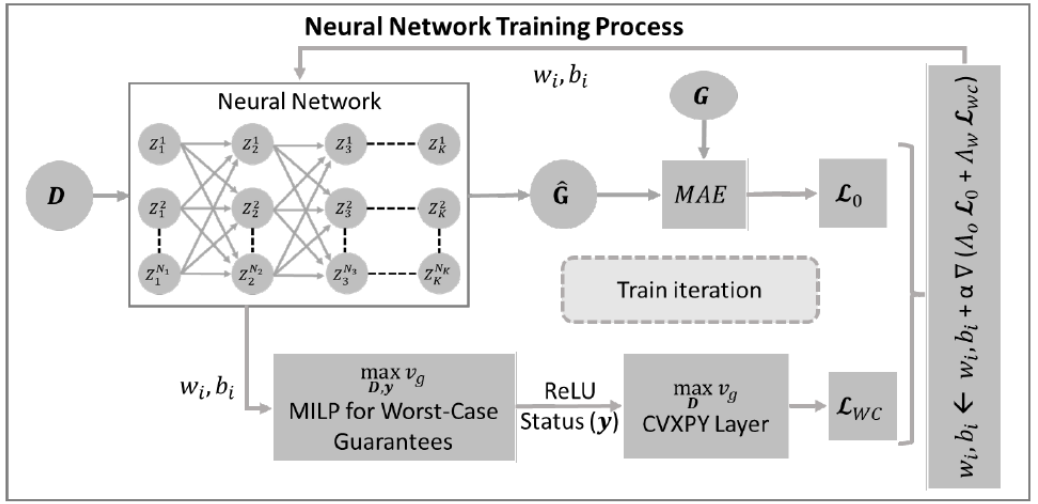
on how to design an NN training that minimizes worst-case violations

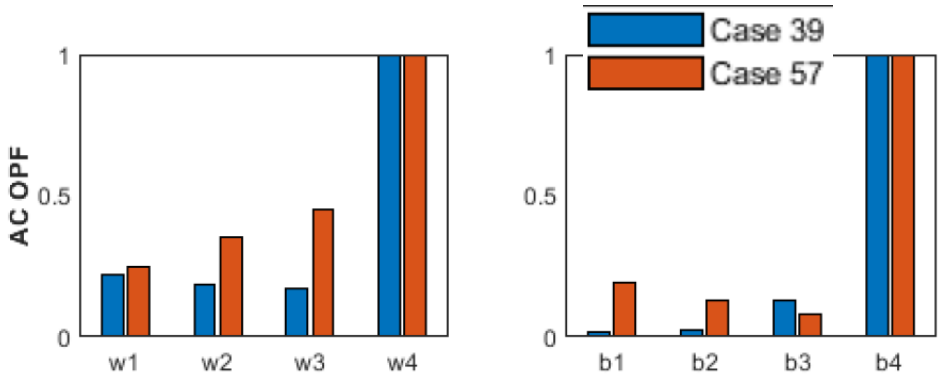
1. Fix the binaries
   - Arbitrary assumption (but it works): for small perturbations of weights & biases, binaries remain constant
   - Solve the lower level MILP by itself, find the binary values for the max constraint violation and fix them

2. MILP is converted to an LP → it is now differentiable

3. Cast it as a differentiable optimization layer (we use CVXPY) → NN training can now backpropagate through it

4. Reduce complexity: reduce #weights and #biases to adjust → **w, b of last layer had the largest impact**





Derivatives of weights and biases on each of the 4 layers w.r.t. worst-case violations

R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf

Test Cases		MAE (%)	Worst-Case Guarantees Max. Generation Violation w.r.t. Max. Loading
case39	NN	AC-OPF	
	GenNN		
	WCNN		
case57	NN		
	GenNN		
	WCNN		
case118	NN		
	GenNN		
	WCNN		
case162	NN		
	GenNN		
	WCNN		

NN: standard NN

GenNN: penalizing violations in the Loss Function

WCNN: our approach; penalizing *worst-case* violations

R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf

Test Cases		MAE (%)	Worst-Case Guarantees Max. Generation Violation w.r.t. Max. Loading
case39	NN	0.56%	0.67%
	GenNN	0.55%	0.67%
	WCNN	0.47%	0.00%
case57	NN	1.02%	0.65%
	GenNN	1.01%	0.67%
	WCNN	1.00%	0.29%
case118	NN	0.42%	204.60%
	GenNN	0.42%	213.80%
	WCNN	0.42%	109.83%
case162	NN	1.10%	184.30%
	GenNN	1.06%	181.52%
	WCNN	1.06%	142.35%

NN: standard NN

GenNN: penalizing violations in the Loss Function

WCNN: our approach; penalizing *worst-case* violations

1. Good average performance and minimum worst-case violations are **not necessarily competing objectives**

2. Surprising: **WCNN** not only eliminates all violations, but manages to find a lower minimum for the average performance as well

R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf

Test Cases		MAE (%)	Worst-Case Guarantees Max. Generation Violation w.r.t. Max. Loading
case39	NN	0.56%	0.67%
	GenNN	0.55%	0.67%
	WCNN	0.47%	0.00%
case57	NN	1.02%	0.65%
	GenNN	1.01%	0.67%
	WCNN	1.00%	0.29%
case118	NN	0.42%	204.60%
	GenNN	0.42%	213.80%
	WCNN	0.42%	109.83%
case162	NN	1.10%	184.30%
	GenNN	1.06%	181.52%
	WCNN	1.06%	142.35%

NN: standard NN

GenNN: penalizing violations in the Loss Function

WCNN: our approach; penalizing *worst-case* violations

1. For larger systems, the worst-case violations are large
2. **WCNN manages to reduce them by 50%**
3. Reducing Worst-Case Violations does not affect average performance!

A lot more work is needed to improve scalability and performance!

R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf

Thoughts on Minimizing Worst-Case Violations of Neural Networks

- What did I show?
  1. Verify the output of a trained NN
  2. Incorporate the ground truth in #1 → determine worst-case violations of a trained NN
  3. Incorporate #2 in NN training →for the first time, create a NN training procedure that can determine and reduce the worst-case violations *during training*

Contributions from our group which can be used in the field of ML too

- Why does it work?
  - Because we have a physical model of the process that our NN emulates

- What are the challenges?
  - Computational performance → it takes too much time
  - Scalability → how can we verify larger neural networks (or consider more complex ground truth representations)
  - How can we always achieve zero MILP gap = obtain the performance guarantee?

One approach for scalability:

S. Chevalier, S. Chatzivasileiadis, *Global Performance Guarantees for Neural Network Models of AC Power Flow*
https://arxiv.org/abs/2211.07125

- Solutions? .....

Integrating Worst-Case Violations in NN training -- End

(Physics-Informed) Graph Neural Networks for Fast N-k Security Assessment --Begin

Agnes Nakiganda, Spyros Chatzivasileiadis, **Graph Neural Networks for Fast Contingency Analysis of Power Systems**, 2025. Online https://arxiv.org/abs/2310.04213

European Research Council
Established by the European Commission

Agnes Nakiganda

Postdoc

Imperial College (formerly with DTU)

Agnes Nakiganda, Spyros Chatzivasileiadis, **Graph Neural Networks for Fast Contingency Analysis of Power Systems**, 2025. Online https://arxiv.org/abs/2310.04213

European Research Council
Established by the European Commission

What is the goal?

- Train a Graph Neural Network to estimate voltages and line flows of N-k contingencies
- Use GNN as a fast screening tool!
- Training only on base topology (N-0) and all N-1 cases
- Estimate line flows and voltages for all N-2 cases and N-3 cases
  - **No N-2 and N-3 cases were used for training**
  - N-2 and N-3 were used only for testing
- Why GNN? Because it captures topology

Why?

TABLE I
CHARACTERISTICS OF THE TEST NETWORKS

Network	6-Bus	24-bus	57-bus	118-bus
Nodes	6	24	57	118
Branches	11	33	63	173
Transformers	0	5	17	13
Generators	2	10	6	53
Loads	3	17	42	99
Eligible N−1 topologies	11	32	62	166
Eligible N−2 topologies	55	505	1'928	14'408
Eligible N−3 topologies	165	4'885	37'765	793'206

- 118-bus → >700'000 N-3 contingencies *for a single generation and demand scenario*
- Assume 19 generators with a high and low generation scenario
- Assume a high and a low demand profile (all loads vary uniformly)
- Total: 1,000,000 scenarios x 700,000 contingencies → we need to assess over 700 billion scenarios…!

What will we talk about?

- **2 Different Graph-Aware Neural Networks**
  - Guided Droupout
  - Edge-Varying Graph Neural Network
- **With and without a Physics-Informed** Loss Term and equations
  - The first to define and investigate a Physics-Informed Guided Dropout Neural Network
  - Among the first to work with Physics-Informed Graph Neural Networks

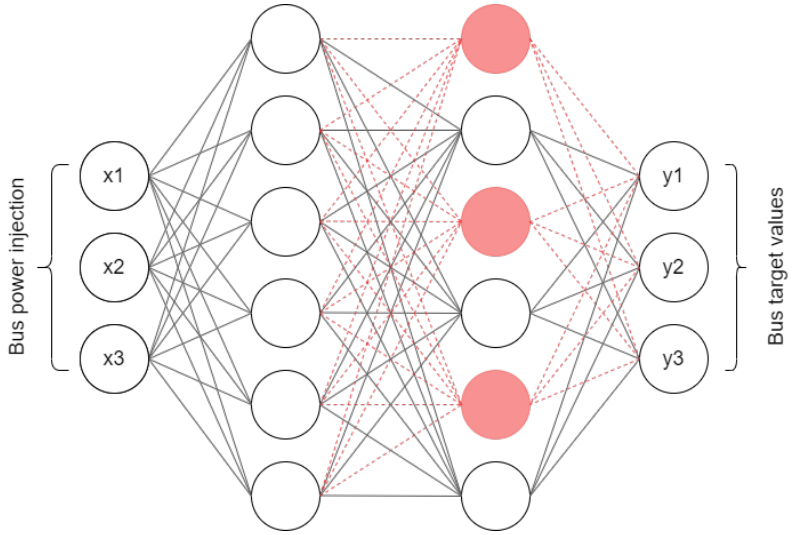
1. Investigate the performance of **4 different Graph-Aware Neural Networks**
   1. **Guided Dropout** without Physics-Informed
   2. **Guided Dropout** with Physics-Informed
   3. Edge-Varying **Graph Neural Network** without Physics-Informed
   4. Edge-Varying **Graph Neural Network** with Physics Informed
2. **Compare** their performance with **DC Power Flow** which is considered a standard tool to assess fast N-k contingencies
3. **Assess** their performance in terms of **time**

Guided-Dropout Neural Network

Ref: B. Donnot, I. Guyon, M. Schoenauer, A. Marot, and P. Panciatici, "Fast power system security analysis with Guided dropout," 2018

Base Case N-0

Conditional Neurons are <u>out</u>

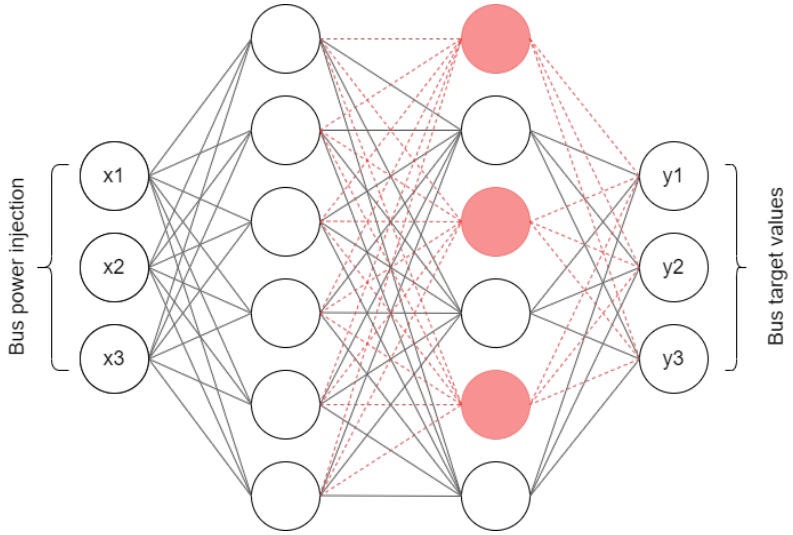


Guided-Dropout Neural Network

Ref: B. Donnot, I. Guyon, M. Schoenauer, A. Marot, and P. Panciatici, "Fast power system security analysis with Guided dropout," 2018

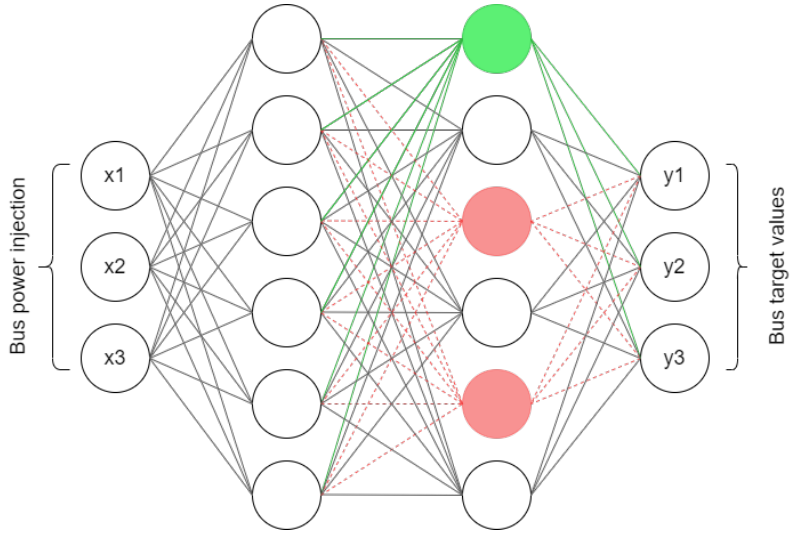
Base Case N-0

Conditional Neurons are **out**



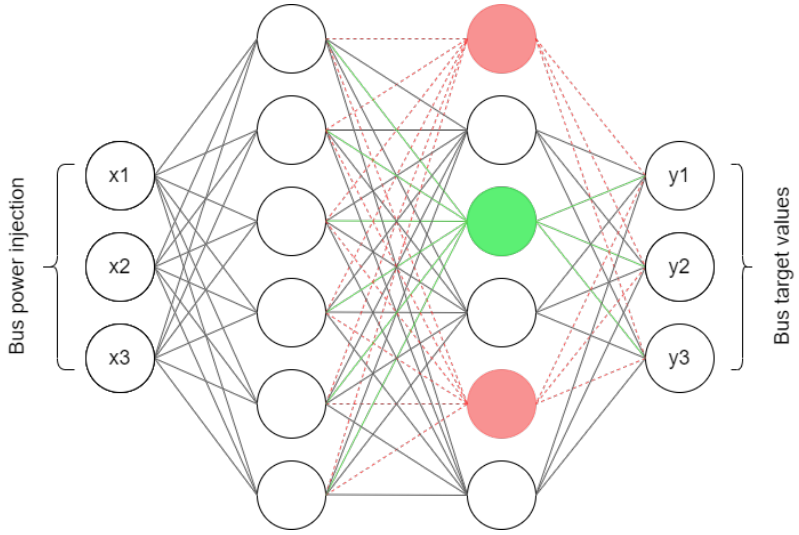
N-1; Line 1 out

Conditional Neuron 1 is **in**



N-1; Line 2 out

Conditional Neuron 2 is **in**

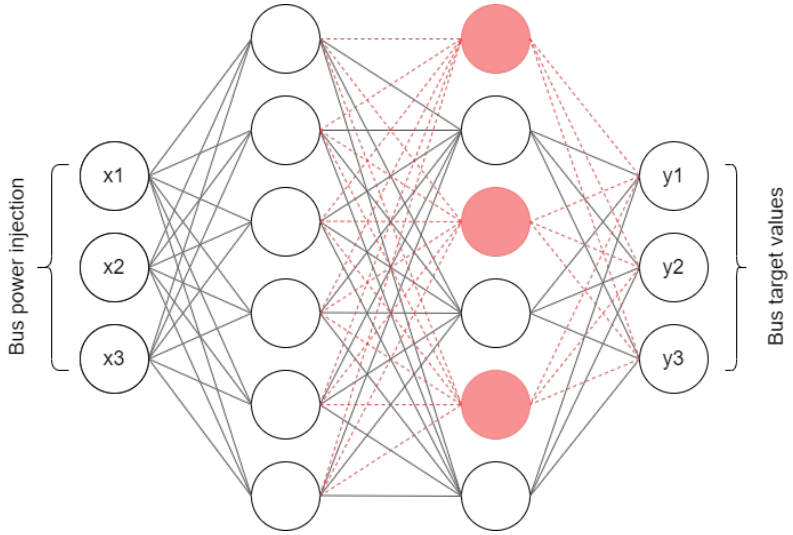


Guided-Dropout Neural Network

Ref: B. Donnot, I. Guyon, M. Schoenauer, A. Marot, and P. Panciatici, "Fast power system security analysis with Guided dropout," 2018

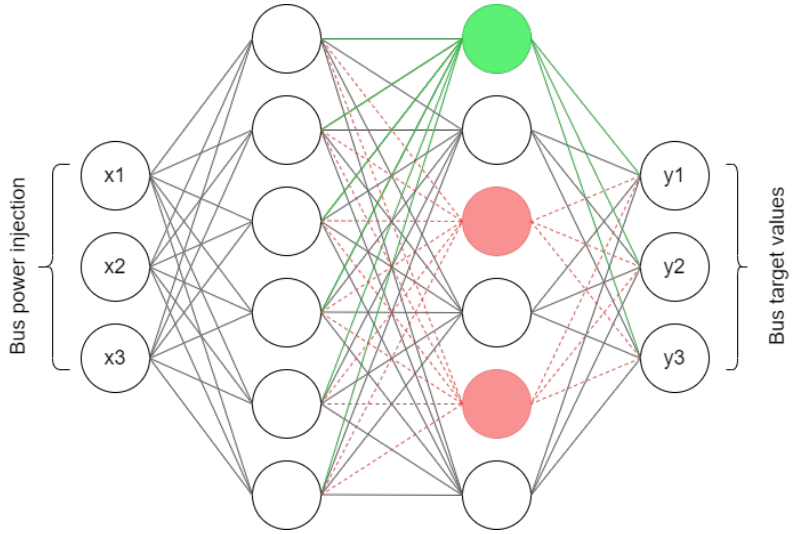
Base Case N-0

Conditional Neurons are **out**



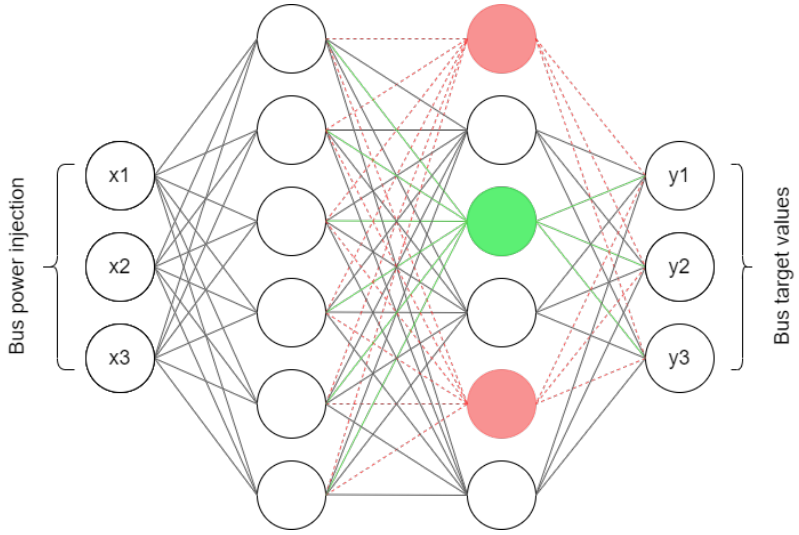
N-1; Line 1 out

Conditional Neuron 1 is **in**



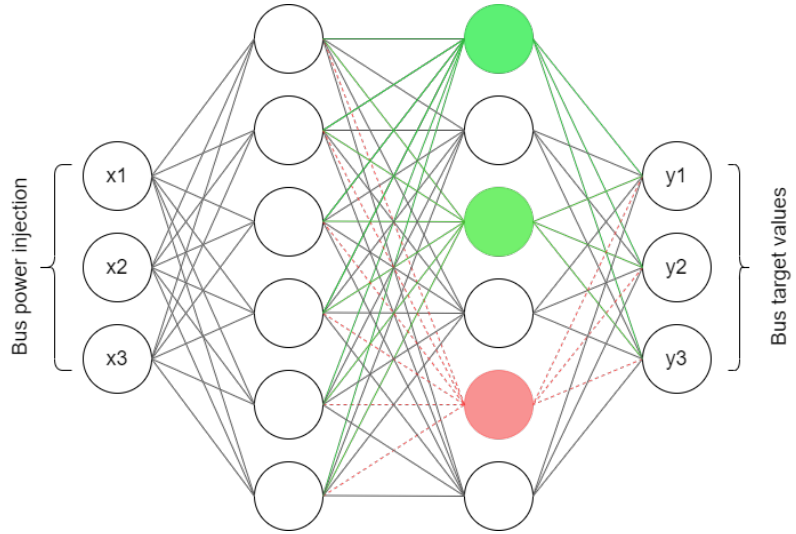
N-1; Line 2 out

Conditional Neuron 2 is **in**



N-2; Lines 1 and 2 are out

Conditional Neurons 1 and 2 are **in**



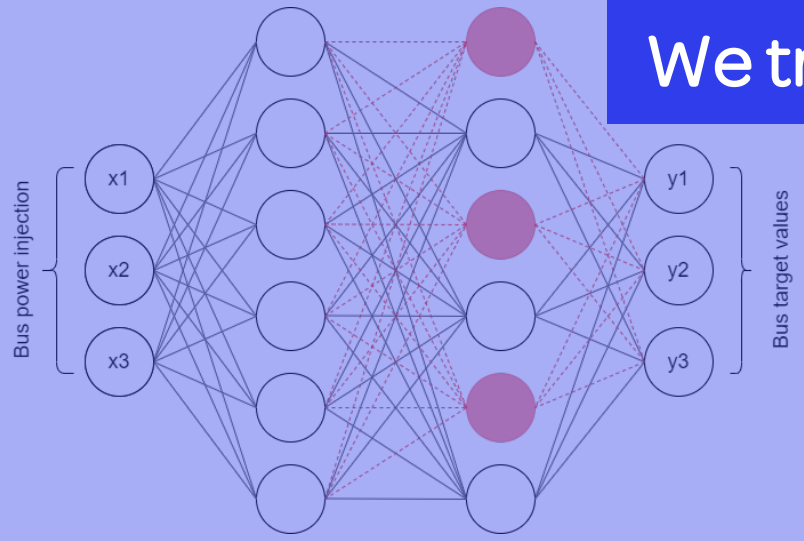
Guided-Dropout Neural Network

Ref: B. Donnot, I. Guyon, M. Schoenauer, A. Marot, and P. Panciatici, "Fast power system security analysis with Guided dropout," 2018

We train for this

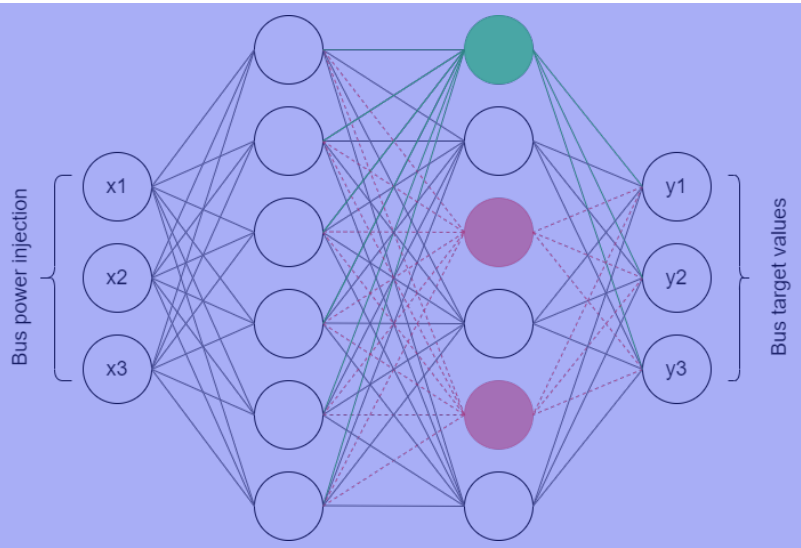
Base Case N-0

Conditional Neurons are **out**



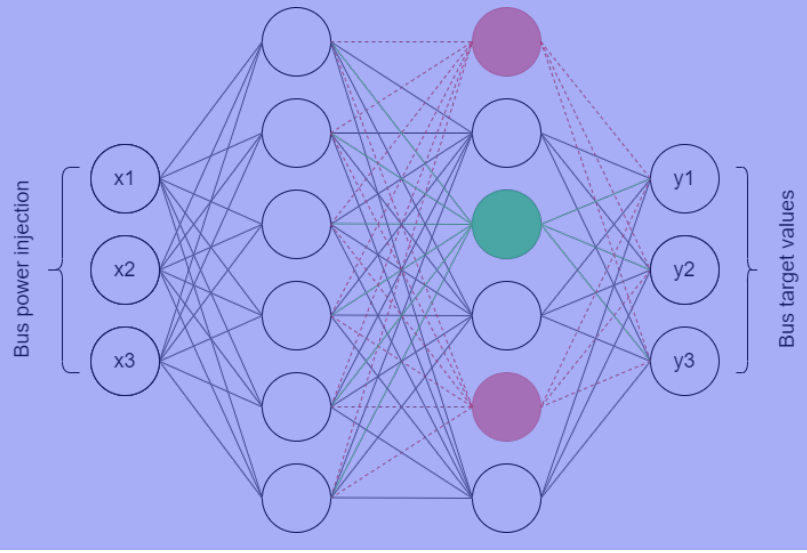
N-1; Line 1 out

Conditional Neuron 1 is **in**



N-1; Line 2 out

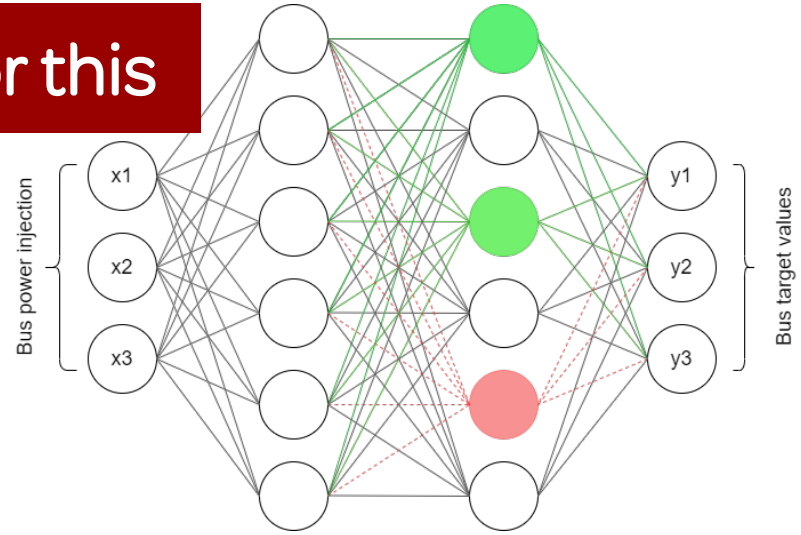
Conditional Neuron 2 is **in**



We test for this

N-2; Lines 1 and 2 are out

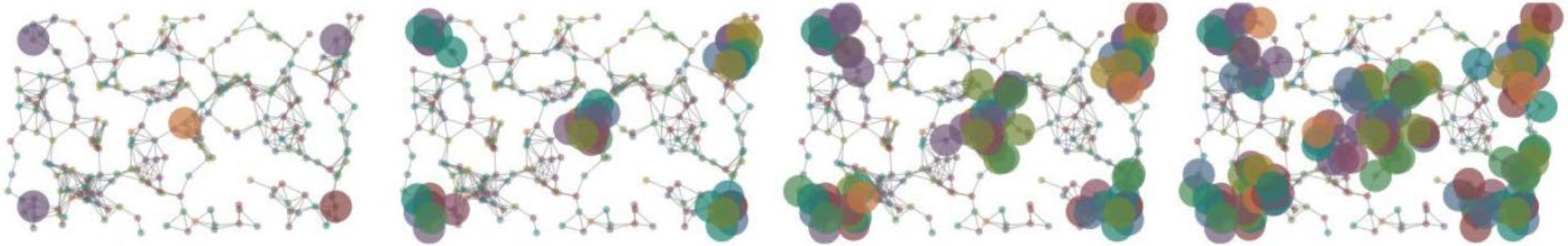
Conditional Neurons 1 and 2 are **in**

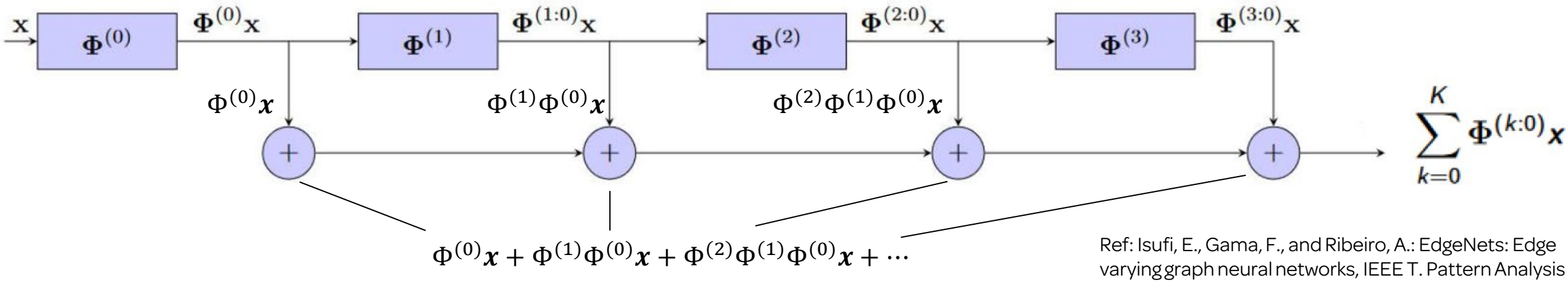


Graph Neural Networks

- $\Phi^{(k)}$ encodes the NN weights based on the graph adjacency matrix → Neurons are connected based on the topology of the network

As we increase the hops, we widen the neighborhood that influences a specific node



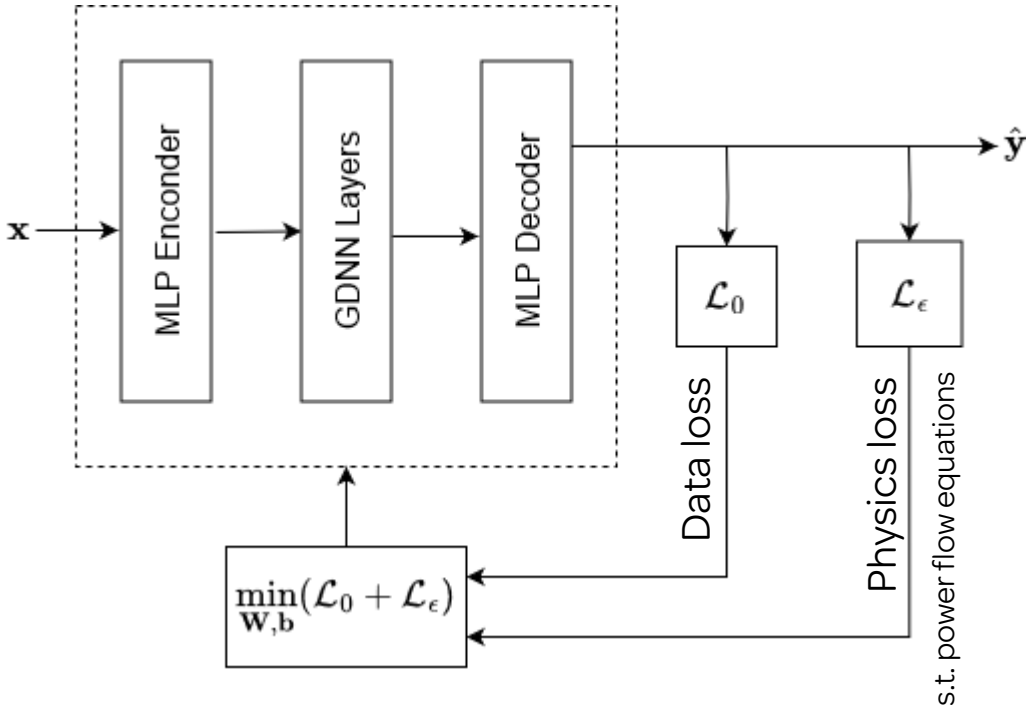


Ref: Isufi, E., Gama, F., and Ribeiro, A.: EdgeNets: Edge varying graph neural networks, IEEE T. Pattern Analysis

Physics Informed Graph-Aware Neural Networks

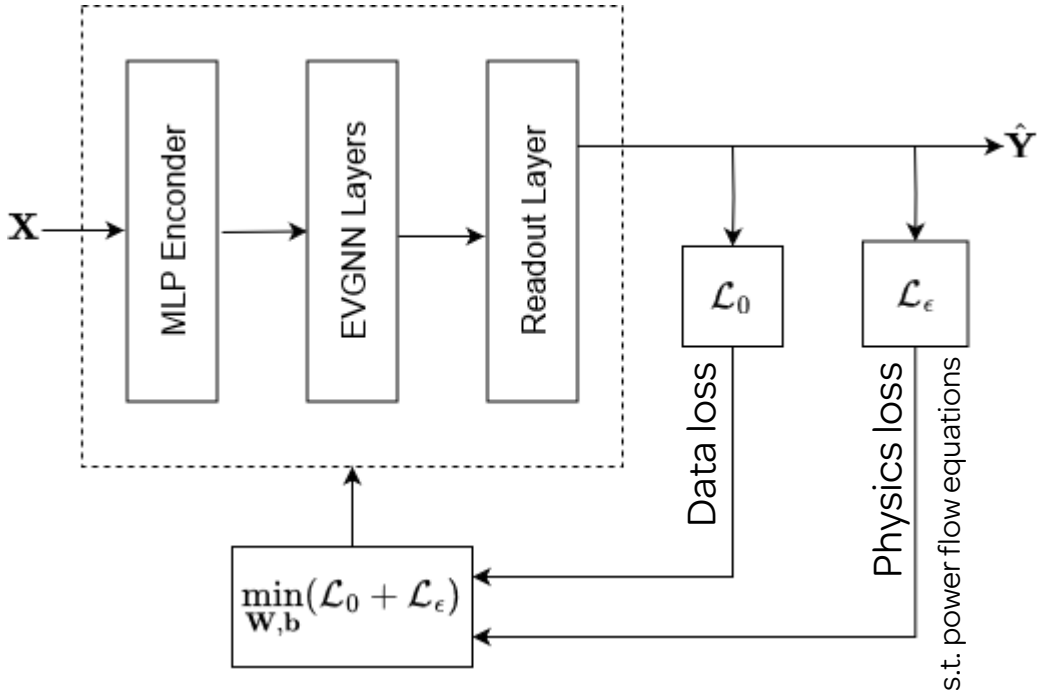
PI-GDNN

Physics-Informed Guided Dropout

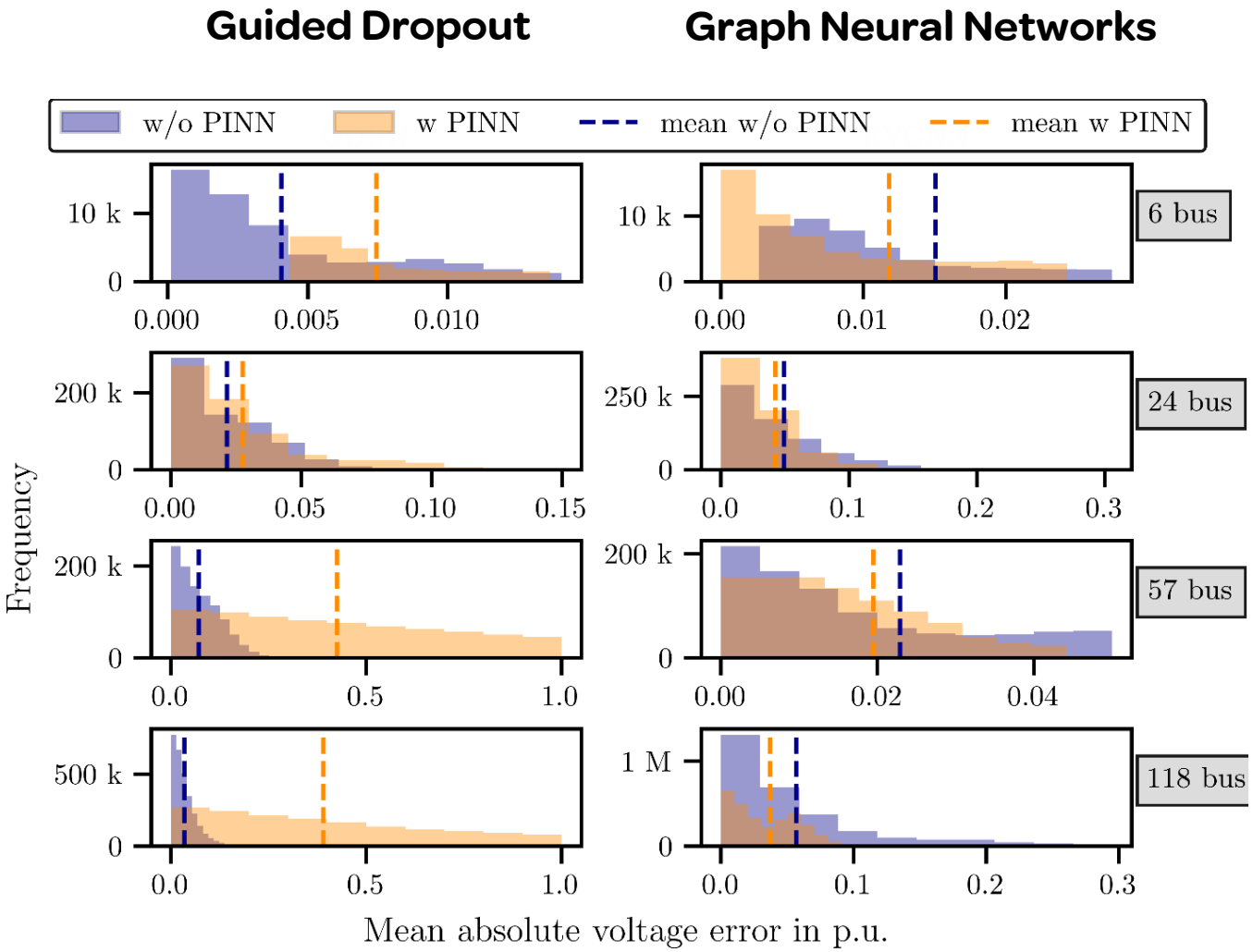


PI-EVGNN

Physics-Informed Graph Neural Network



Physics-Informed NNs do not always perform better

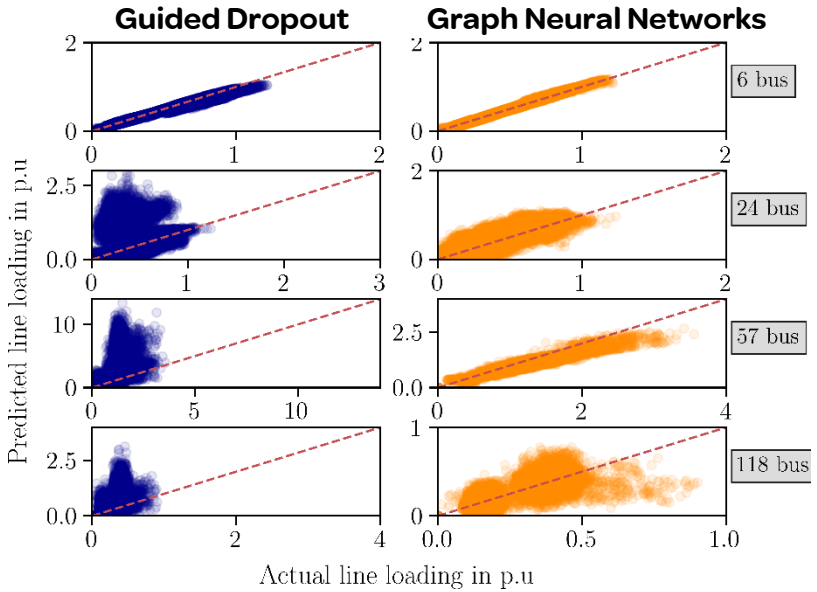


PINNs vs non-PINNs

- **Physics-Informed Graph Neural Networks perform better** than non-Physics-Informed
- **Non-Physics-Informed Guided Dropout perform better** than Physics-Informed Guided Dropout
- For the rest of our comparisons, we limit ourselves to 2 models:
  - GDNN
  - PI-EVGNN

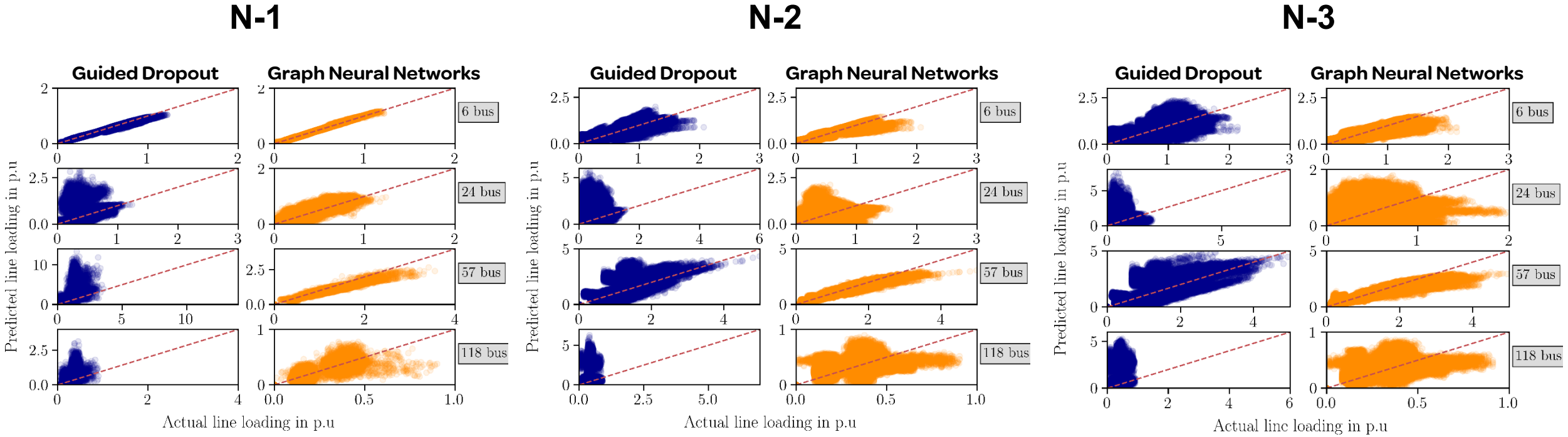
GNNs for Regression: Estimating the line flows

N-1



GNNs for Regression: Estimating the line flows

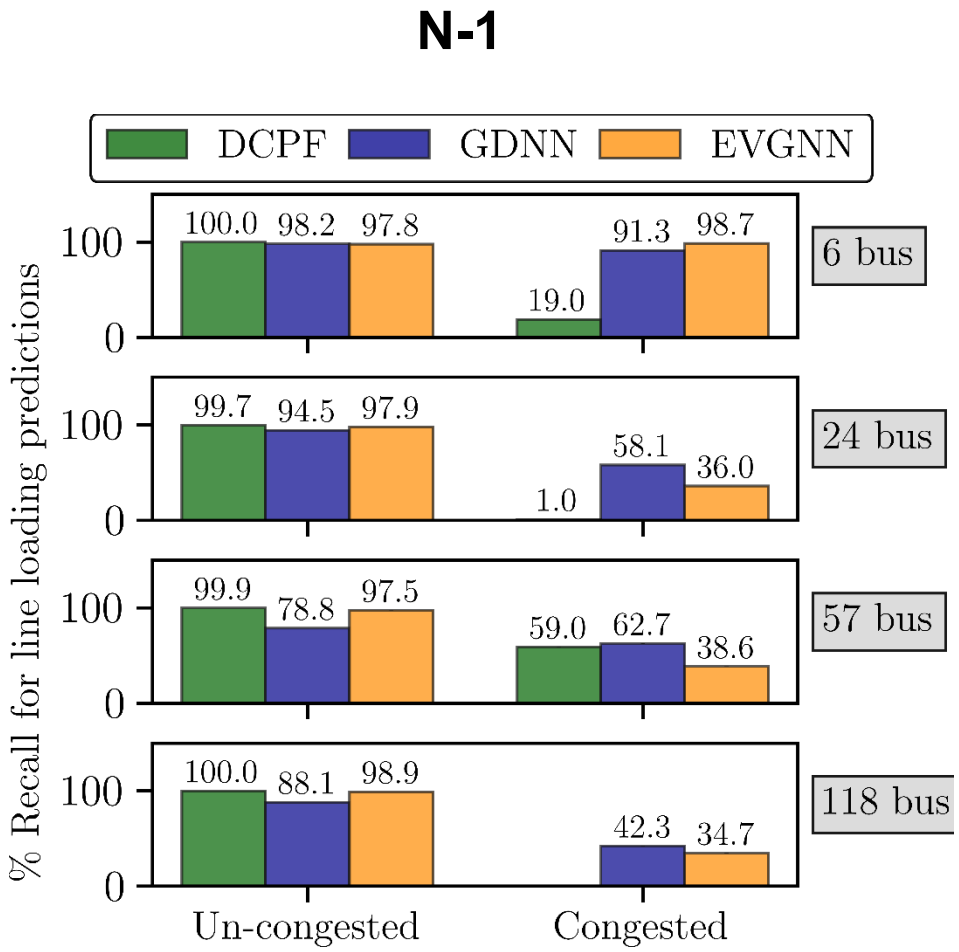
No training on N-2 and N-3, only testing!



- Estimating the bus voltages had in general a **better** performance from the line flows

More info here: Agnes Nakiganda, Spyros Chatzivasileiadis, **Graph Neural Networks for Fast Contingency Analysis of Power Systems**, 2025. Online https://arxiv.org/abs/2310.04213

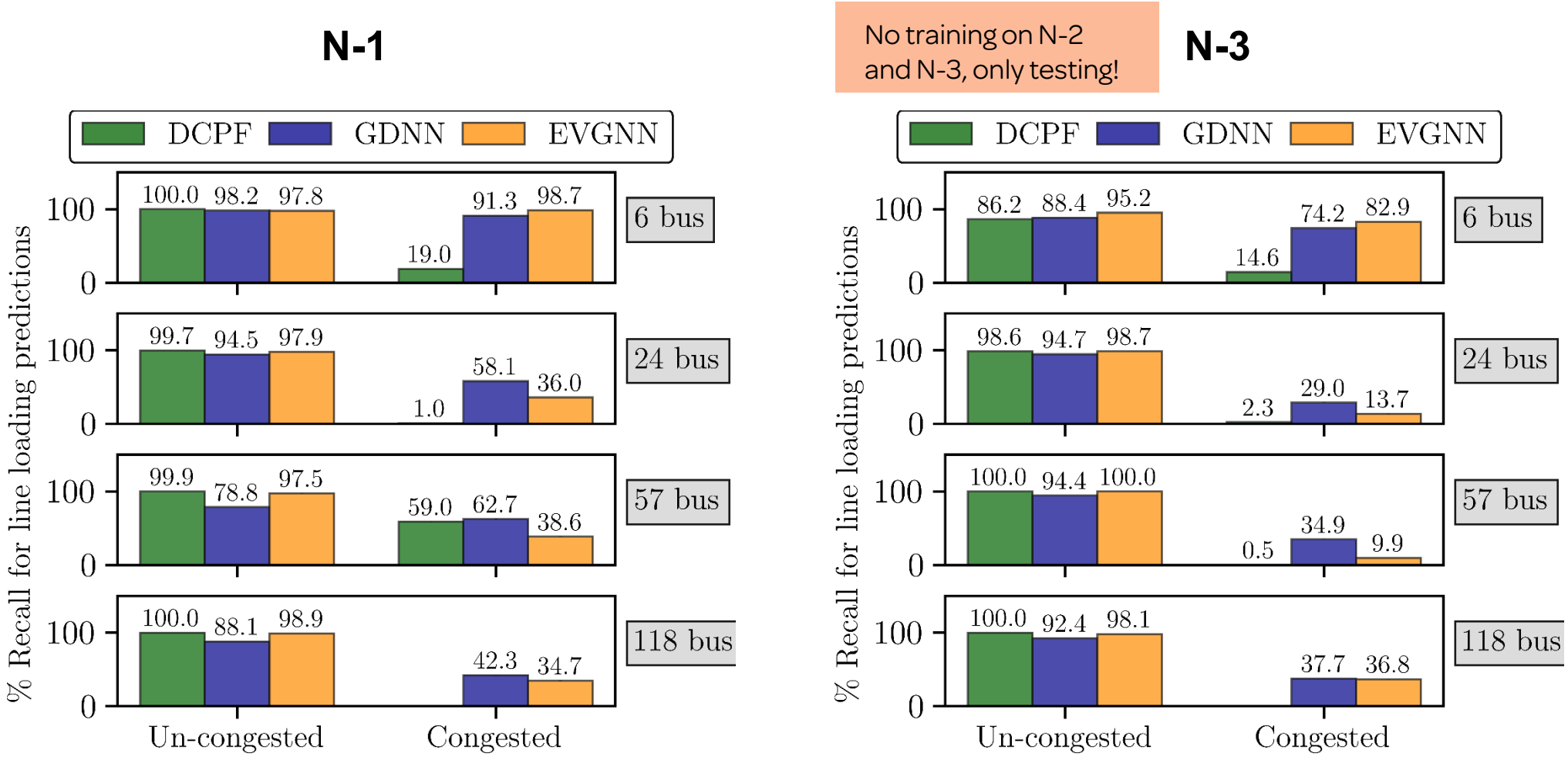
GNNs vs DC Power Flow: Estimating Line Overloadings



Metric: Recall (True Positive Rate) ; Recall=100%: NN has classified correctly all data points belonging to a class

GNNs vs DC Power Flow: Estimating Line Overloadings

No training on N-2 and N-3, only testing!

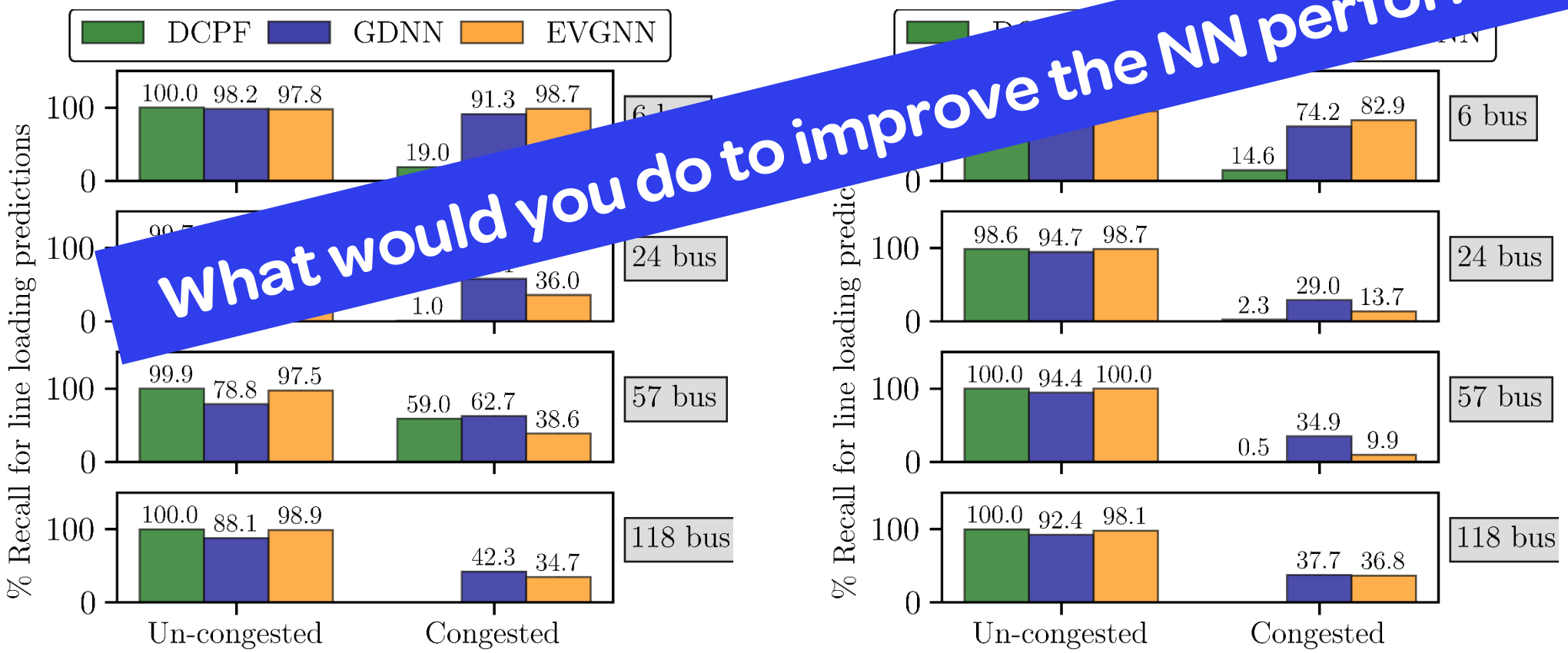


- DC Power Flow performs the worst: cannot estimate any line congestion
- Both Guided Dropout and Graph NN perform better, but not much better

Metric: Recall (True Positive Rate) ; Recall=100%: NN has classified correctly all data points belonging to a class

No training on N-2 and N-3, only testing!

What would you do to improve the NN performance?



- DC Power Flow performs the worst: cannot estimate any line congestion
- Both Guided Dropout and Graph NN perform better, but not much better

Metric: Recall (True Positive Rate) ; Recall=100%: NN has classified correctly all data points belonging to a class

What would you do to improve the NN performance?

- We need better databases!
- And better methods to generate these databases fast and with information-rich content!
- Some first efforts from our side:

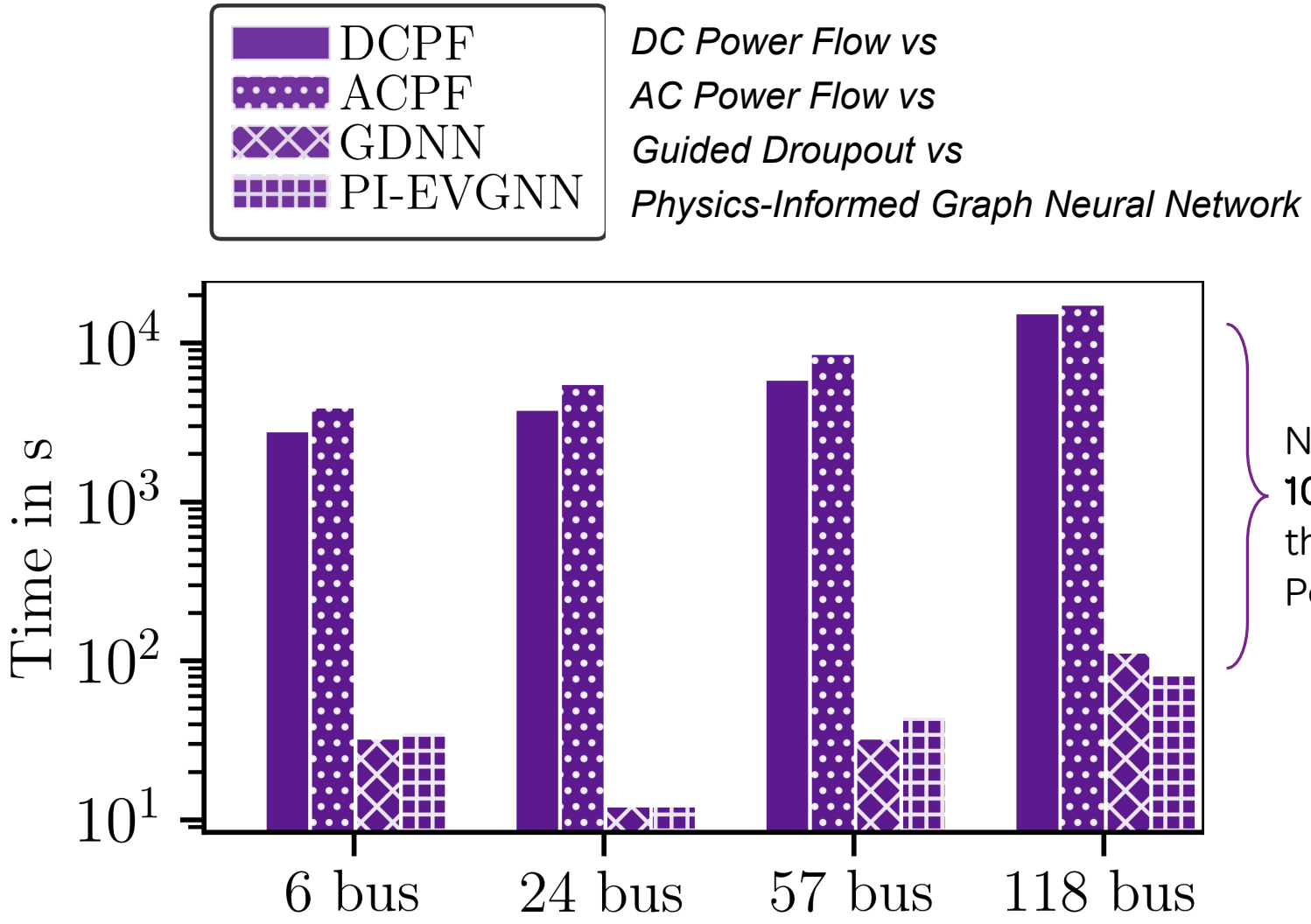
F. Thams, A. Venzke, R. Eriksson, S. Chatzivasileiadis. Efficient Database Generation for Data-Driven Security Assessment of Power Systems. *IEEE Transactions on Power Systems*, vol 35, no. 1, pp. 30-41, Jan. 2020
[.pdf | Databases | IEEEXplore]

Bastien Giraud, Lola Charles, Agnes Marjorie Nakiganda, Johanna Vorwerk, Spyros Chatzivasileiadis, A Dataset Generation Toolbox for Dynamic Security Assessment: On the Role of the Security Boundary, IREP 2025, https://arxiv.org/abs/2501.09513

Open-source toolbox!

Which method you think is the fastest ?

Evaluation time



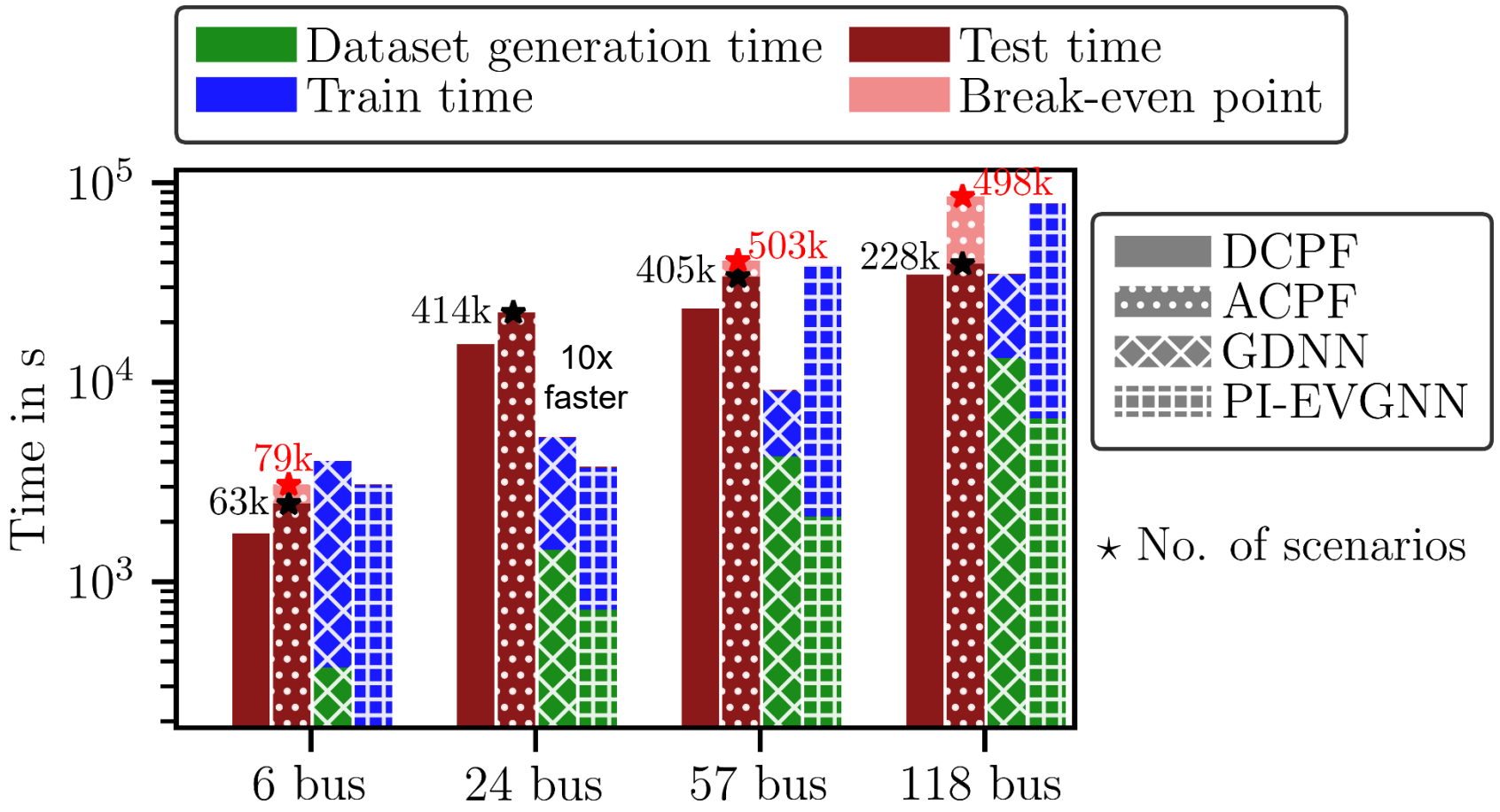
- Logarithmic Axis!

Neural Networks
100-400 times faster
than AC and DC
Power Flow

- NNs need **1.5 minutes** to assess 100,000 scenarios
- AC/DC Power Flow need **5 hours** to assess 100,000 scenarios

What happens if we include the training time?

Computation Time including NN training



- **Logarithmic Axis!! Bar length not proportional to time**
- For the larger systems, it appears that the break-even point is at approx. 500,000 scenarios
  - For more than 500,000 scenarios the NNs are faster
- Considering that we talked about 700 billion scenarios (118-bus, N-3 cases), then NNs appear very promising for screening

Conclusions

- Power systems need Trustworthy AI!

- **Graph-Aware Neural Networks are a promising option** to screen a vast number of N-k contingences (hundreds of millions)
  - Can capture topology changes
  - **Can be 100x-400x faster** in their evaluation (1.5 minutes instead of 5 hours for 100,000 scenarios)
  - Much better performance than DC Power Flow

- Including training time, the break-even point with conventional methods appears to be at over 500,000 scenarios (57-bus, 118-bus)
  - Considering that a moderate assessment of N-3 contingencies in the 118-bus system might require 700 billion scenarios, the break-even point is low

- **But: The screening performance still needs to be improved**. A lot of R&D potential in:
  - Efficient and information-rich database generation for NN training
  - Improved NN training, e.g. design of input and output vectors, NN structures
  - Inclusion of Physics-informed terms or not

(Physics-Informed) Graph Neural Networks for Fast N-k Security Assessment
--End

Agnes Nakiganda, Spyros Chatzivasileiadis, **Graph Neural Networks for Fast Contingency Analysis of Power Systems**, 2025. Online https://arxiv.org/abs/2310.04213

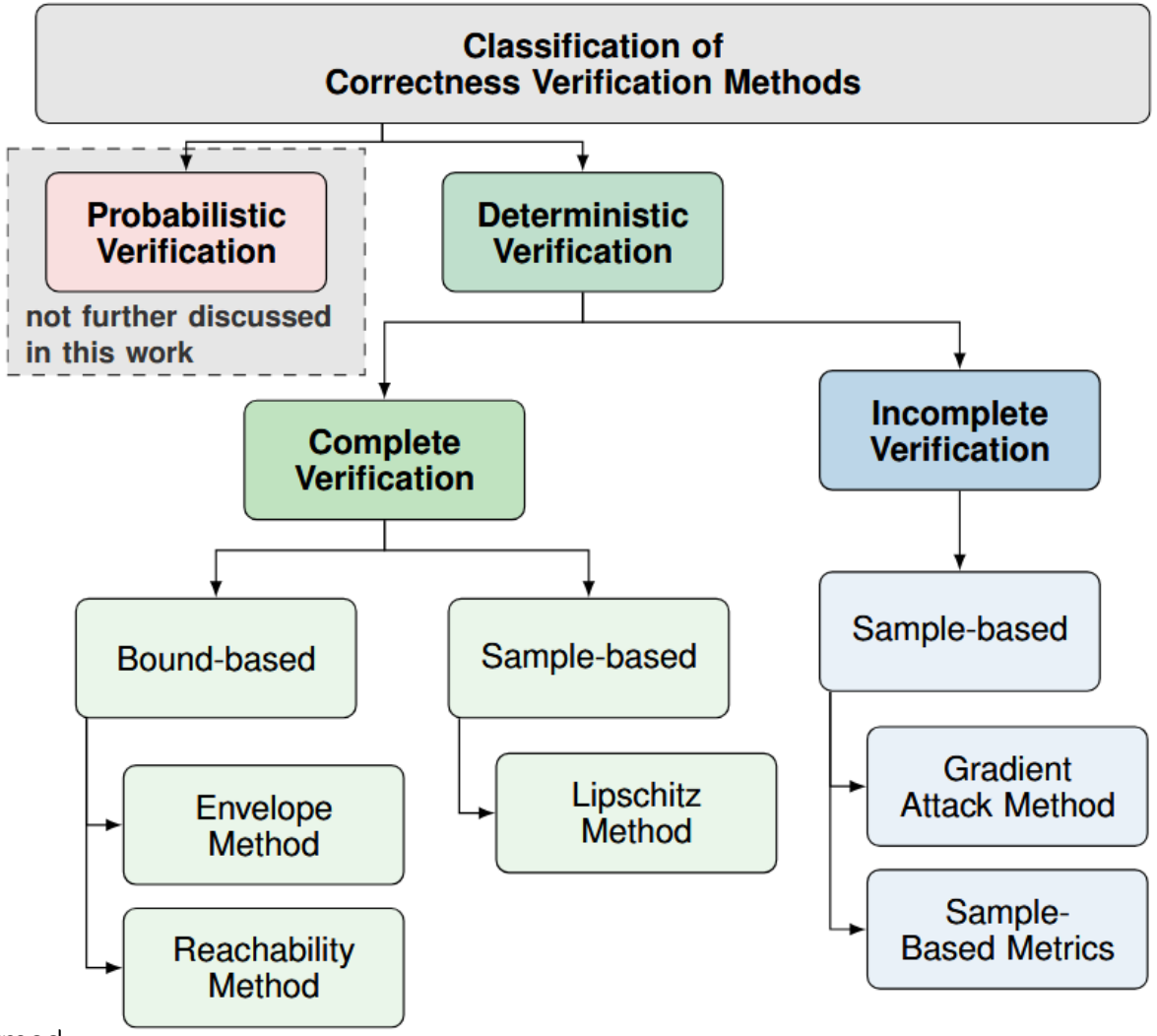
European Research Council
Established by the European Commission

Conclude and Discuss --Begin

European Research Council
Established by the European Commission

Classification of Verification Methods

Open-Source Trustworthiness Toolbox coming soon!



P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. https://doi.org/10.1016/j.segan.2025.101818

Trustworthy AI for Power Systems: Vision

AI Testing and Experimentation Facility for Energy

- Establish a platform that verifies AI tools and certifies that they comply with power system safety specifications

AI-EFFECT EU project

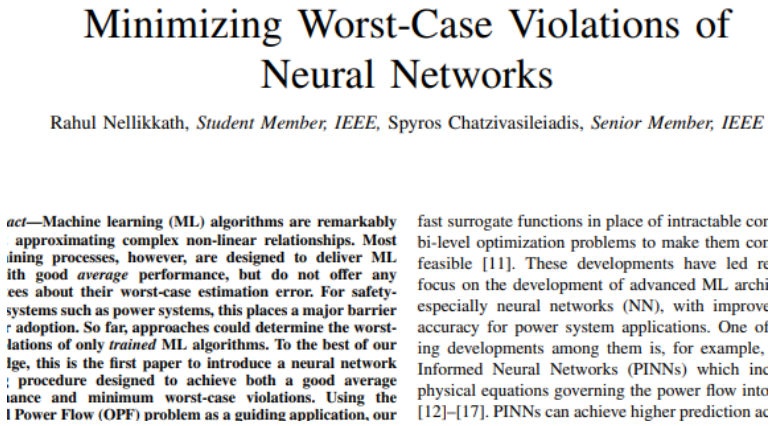
Start: 1st October 2024

Participants: EPRI (Lead), DTU, TU Delft, Univ. Porto, BEOF, TenneT, ENEL, and others

AI Standards: Create Standards for AI tools in Energy

Design a Neural Network Training Algorithm that simultaneously delivers guarantees of the worst-case NN performance

- Example: "Neural Network Training finished. Accuracy 99.2%. Worst-case violation of critical constraints: 10%."



Minimizing Worst-Case Violations of Neural Networks

Rahul Nellikkath, *Student Member, IEEE,* Spyros Chatzivasileiadis, *Senior Member, IEEE*

act—Machine learning (ML) algorithms are remarkably approximating complex non-linear relationships. Most ining processes, however, are designed to deliver ML ith good *average* performance, but do not offer any ees about their worst-case estimation error. For safety- systems such as power systems, this places a major barrier adoption. So far, approaches could determine the worst- lations of only *trained* ML algorithms. To the best of our lge, this is the first paper to introduce a neural network procedure designed to achieve both a good average iance and minimum worst-case violations. Using the l Power Flow (OPF) problem as a guiding application, our

fast surrogate functions in place of intractable cor bi-level optimization problems to make them cor feasible [11]. These developments have led re focus on the development of advanced ML archi especially neural networks (NN), with improve accuracy for power system applications. One of ing developments among them is, for example, Informed Neural Networks (PINNs) which inc physical equations governing the power flow into [12]–[17]. PINNs can achieve higher prediction ac

R.Nellikkath, S. Chatzivasileiadis, Minimizing worst-case violations for neural networks, https://arxiv.org/abs/2212.10930

Some Final Thoughts

- **If we want to accelerate processes by 10x-100x-1000x we need to think differently**
  - Conventional methods reach their limits (?)
  - Could Machine Learning become the disruptive technology?

- **AI Needs OR → Neural Network Verification is an optimization problem**. Can we address its challenges?
  - **If yes, we remove barriers** for a wide range of safety-critical applications
    - Power systems, robots, self-driving cars, control of critical infrastructure, and many others

Some Final Thoughts

- **If we want to accelerate processes by 10x-100x-1000x we need to think differently**
  - Conventional methods reach their limits (?)
  - Could Machine Learning become the disruptive technology?

- **AI Needs OR → Neural Network Verification is an optimization problem**. Can we address its challenges?
  - **If yes, we remove barriers** for a wide range of safety-critical applications
    - Power systems, robots, self-driving cars, control of critical infrastructure, and many others

- **Can we model the ground truth? If yes, use it!**
  - Physics-Informed Neural Networks (PINNs)
  - Sampling Beyond Statistics
  - Neural Network Training with Worst-Case Performance Guarantees

- **PINNSim**: A Simulator based on Physics Informed Neural Networks for Power System Dynamics
  - Do not need a single NN for the whole problem
  - **Let's work with "Libraries of Neural Networks"**, similar to "Libraries of Models"
  - A PINN-based simulator can be 10x-100x faster for power system dynamics

Major Challenges

1. SCALABILITY
2. TOPOLOGY
3. How do we take advantage of GenAI?

Open-source Toolboxes

1. Generate your own training datasets!
   GitHub/bastiengiraud/DSA-learn

2. Train your own Physics-Informed Neural Networks!
   GitHub/radiakos/PowerPINN

3. Play with a PINN for Power System Dynamics!
   Google Colab PINN Playground

Thank you!

Spyros Chatzivasileiadis

Professor

www.chatziva.com

spchatz@dtu.dk

References (from our group)

1. B. Giraud , L. Charles, A. M. Nakiganda, J. Vorwerk, S. Chatzivasileiadis, A Dataset Generation Toolbox for Dynamic Security Assessment: On the Role of the Security Boundary, Sustainable Energy, Grids, and Networks, Elsevier, 2025, https://arxiv.org/pdf/2501.09513
2. A. Venzke, D.K. Molzahn, S. Chatzivasileiadis, Efficient Creation of Datasets for Data-Driven Power System Applications. PSCC 2020. https://arxiv.org/pdf/1910.01794.pdf
3. F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient database generation for data-driven security assessment of power systems". ". IEEE Trans. Power Systems, vol. 35, no. 1, pp. 30-41, Jan. 2020. https://www.arxiv.org/abs/1806.0107.pdf
4. R. Nellikkath, S. Chatzivasileiadis. Minimizing Worst-Case Violations of Neural Networks. https://arxiv.org/pdf/2212.10930.pdf
5. Venzke, G. Qu, S. Low, S. Chatzivasileiadis, Learning Optimal Power Flow: Worst-case Guarantees for Neural Networks. **Best Student Paper Award** at IEEE SmartGridComm 2020. https://arxiv.org/pdf/2006.11029.pdf
6. R. Nellikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks for Minimising Worst-Case Violations in DC Optimal Power Flow. In IEEE SmartGridComm 2021, Aachen, Germany, October 2021.
7. R. Nellikkath, S. Chatzivasileiadis. Physics-Informed Neural Networks for AC Optimal Power Flow. PSCC 2020.
8. S. Chevalier, I. Murzakhanov, S. Chatzivasileiadis, *GPU-Accelerated Verification of Machine Learning Models for Power Systems,* **Best Paper Award at HICSS** (Hawaii International Conferences on Systems Sciences), Jan. 2024 https://arxiv.org/pdf/2306.10617
9. P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. https://doi.org/10.1016/j.segan.2025.101818
10. J. Stiasny, S. Chevalier, R. Nellikkath, B. Sævarsson, S. Chatzivasileiadis. Closing the Loop: A Framework for Trustworthy Machine Learning in Power Systems. Accepted to *2022 iREP Symposium - Bulk Power System Dynamics and Control - XI (iREP)*. Banff, Canada. July 2022. [paper | code]
11. A. Venzke, S. Chatzivasileiadis. Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications. *IEEE Transactions on Smart Grid*, Jan. 2021. https://arxiv.org/pdf/1910.01624.pdf
12. R. Nellikkath, A. Venzke, M. K. Bakhshizadeh, I. Murzakhanov, S. Chatzivasileiadis, Physics–Informed Neural Networks for Phase Locked Loop Transient Stability Assessment, PSCC 2024 [https://arxiv.org/abs/2303.12116]
13. G. S. Misyris, A. Venzke, S. Chatzivasileiadis, Physics-Informed Neural Networks for Power Systems. Presented at the Best Paper Session of IEEE PES GM 2020. https://arxiv.org/pdf/1911.03737.pdf
14. I. Karampinis, P. Ellinas, I. Ventura-Nadal, R. Nellikkath, S. Chatzivasileiadis, A Toolbox for Physics-Informed Neural Networks in Power Systems, IEEE Powertech 2025, https://arxiv.org/pdf/2502.06412
15. I. Ventura-Nadal, R. Nelikkath, S. Chatzivasileiadis, Physics-Informed Neural Networks in Power System Dynamics: Improving Simulation Accuracy, IEEE Powertech 2025, https://arxiv.org/pdf/2501.17621