

Trustworthy AI for Power Systems

Spyros Chatzivasileiadis

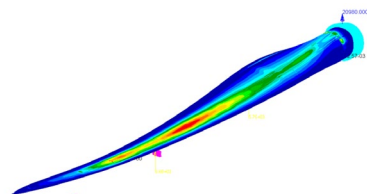
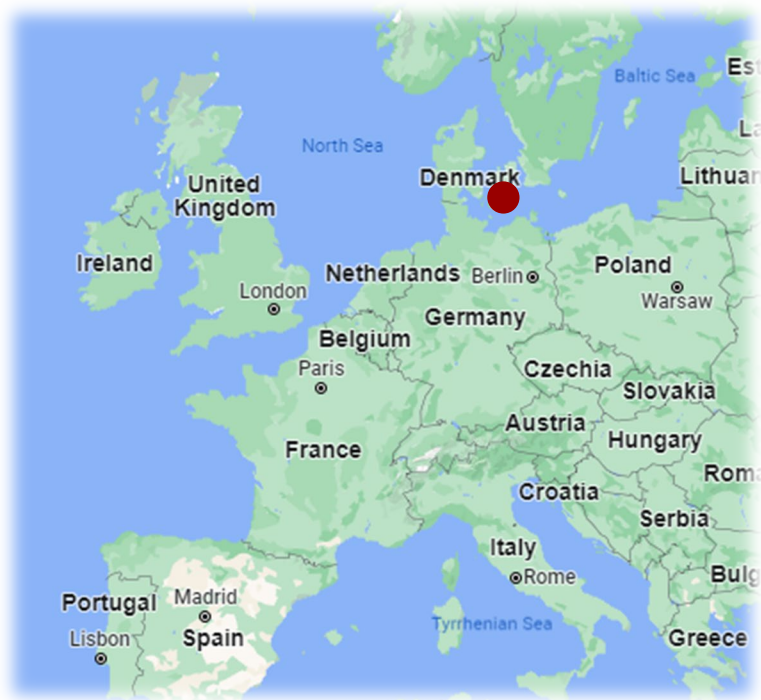
Professor

DTU Wind

Technical Univ. of Denmark

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



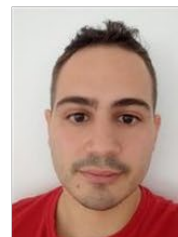
Lejla
Halilbasic



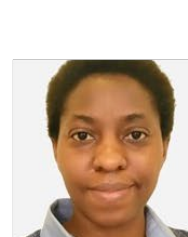
Elea
Prat



Ilgiz
Murzakhanov



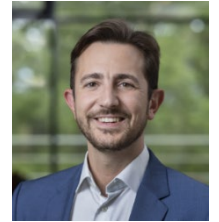
Petros
Ellinas



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, GeorgiaTech
Mathieu Tanneau, GeorgiaTech
Dan Molzahn, GeorgiaTech*

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

Baosen Zhang, Univ. Washington

Kazem Bakhshizadeh, Ørsted

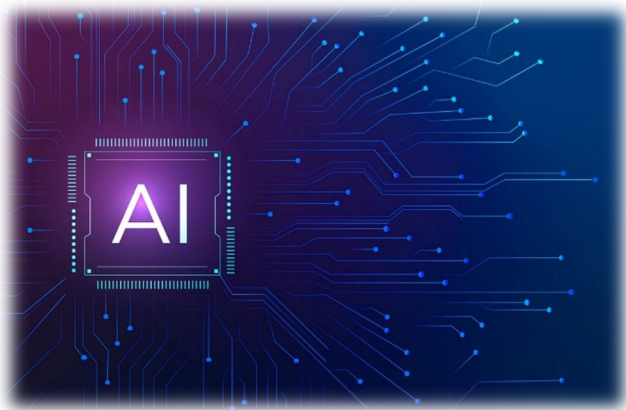


And many thanks to the European Research Council for funding this research



European Research Council

Established by the European Commission



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



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

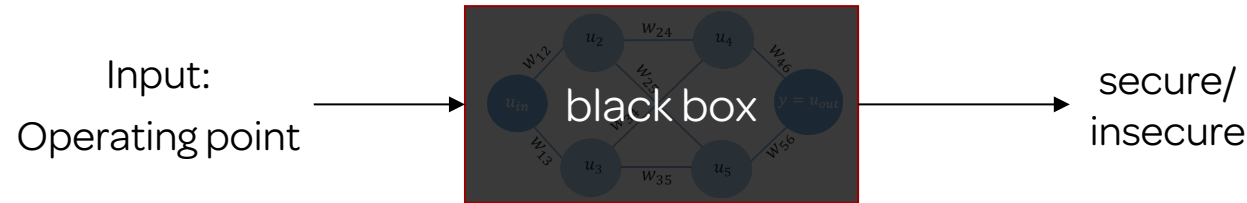
And many more



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

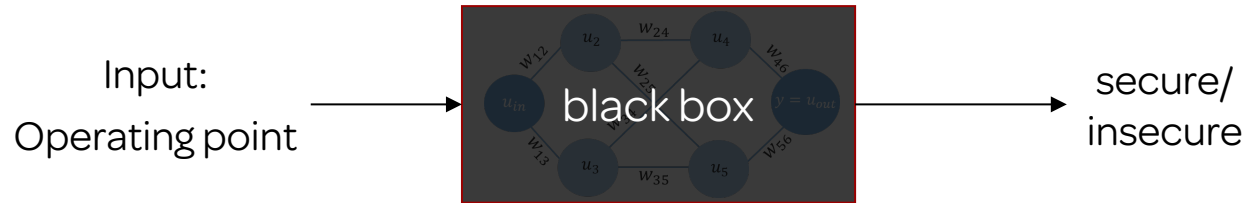


Machine Learning (ML) Barriers for Power systems



1. Why would we use a “**black box**” to decide about a **safety-critical application**?
2. Neural Networks performance metric is “Accuracy”.
Accuracy is a purely statistical performance metric.
Who guarantees that the Neural Network can handle well previously unseen operating points?
3. Good AI Tools need good data. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

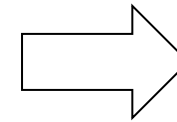
Machine Learning (ML) Barriers for Power systems



1. Why would we use a “**black box**” to decide about a **safety-critical application**?
2. Neural Networks performance metric is “Accuracy”. **Accuracy is a purely statistical** performance metric. Who guarantees that the Neural Network can handle well previously unseen operating points?
3. Good AI Tools need good data. Why would we depend on **discrete and incomplete data**, when we have developed **detailed physical models** over the past 100 years?

Trustworthy AI

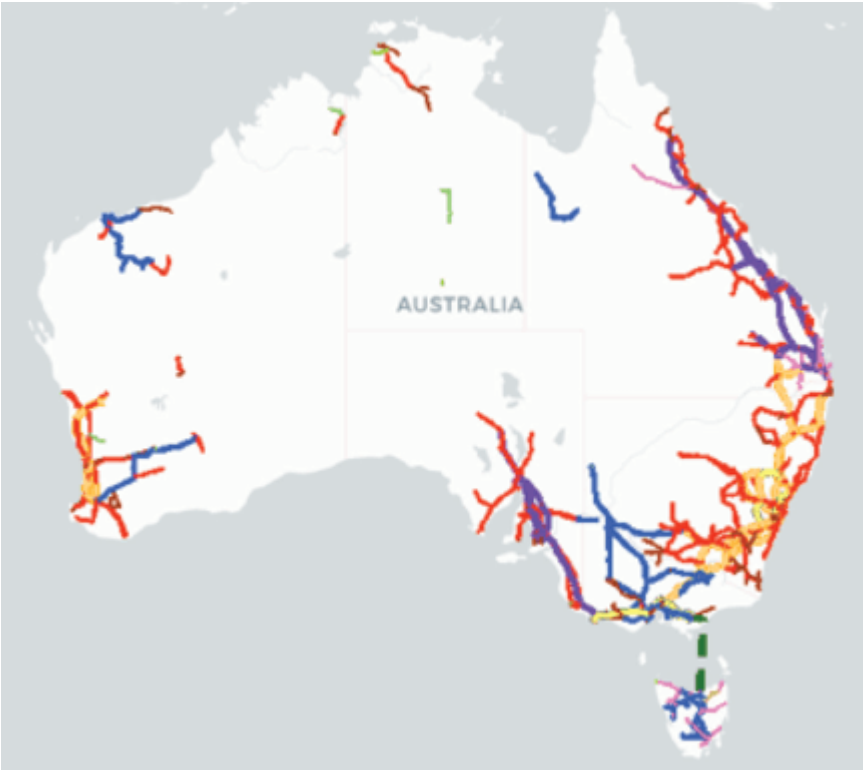
Neural Network verification:
guarantees for the NN performance!



Physics-Informed Neural Networks:
potential to deliver tools that are 10x-100x-1000x faster!

Physics-Informed Neural Networks for Power Systems

What is the challenge?

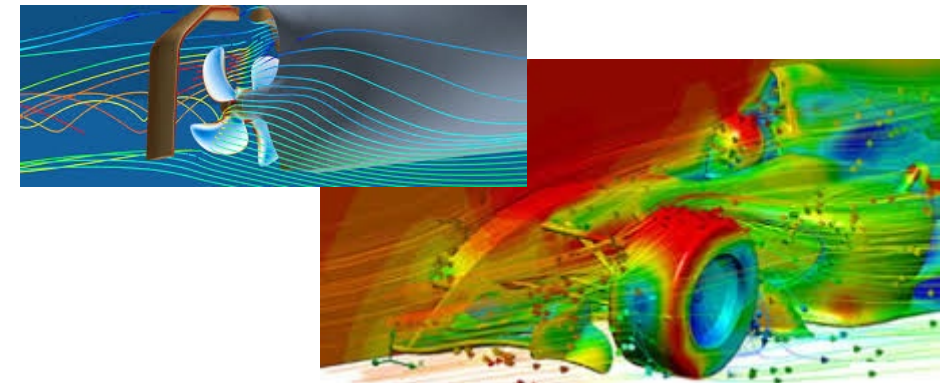
- Assume we operate the **Australian Grid** and need to eliminate the blackout risk for the next day.
 - We need simulations to assess the risk and devise mitigation strategies.
 - **Simulating 20 seconds of the dynamic behavior of the Australian Grid requires 12 minutes** with a current state-of-the-art 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 is to 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)



Physics-Informed Neural Networks for Power Systems

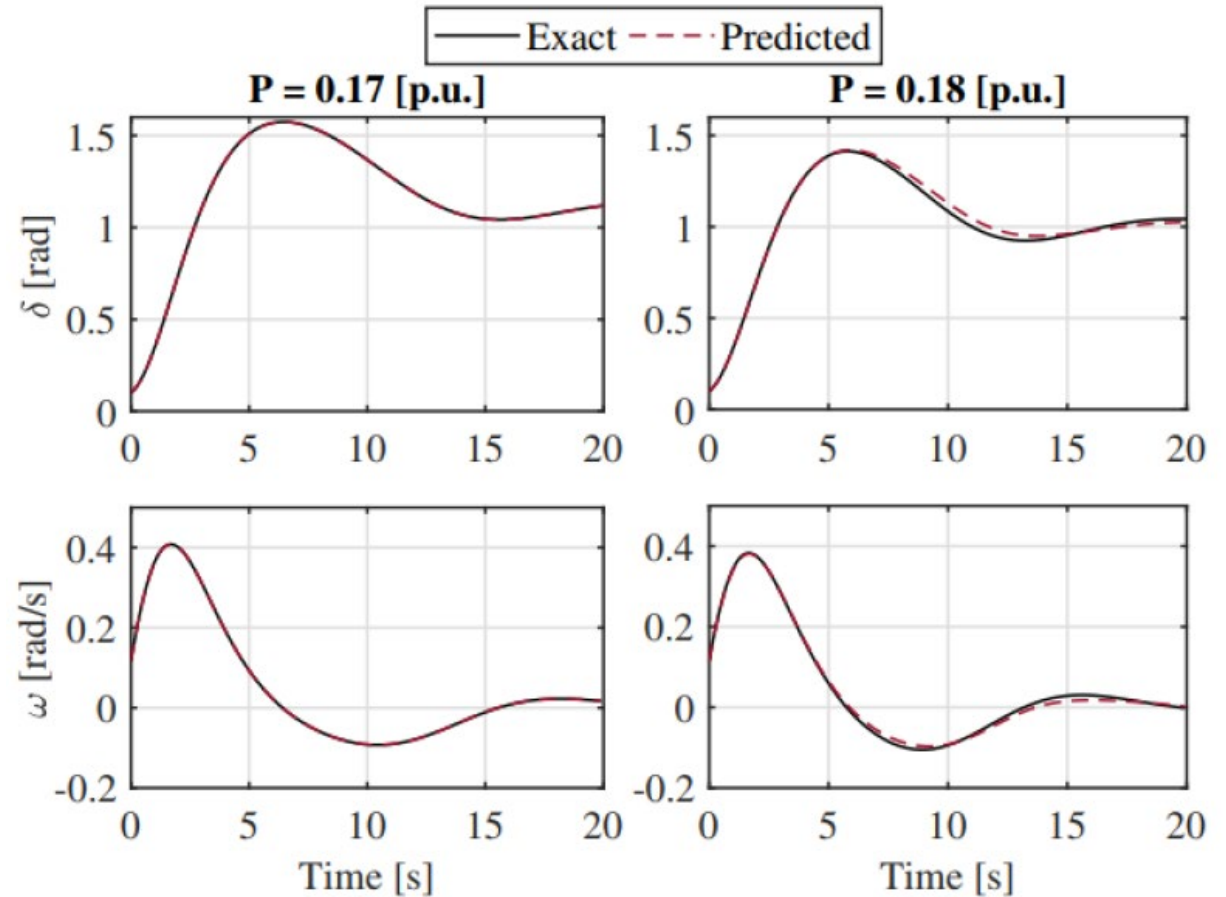
“Original”
Loss function

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

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

$$\dot{\hat{\delta}} = \frac{\partial \hat{\delta}}{\partial t}, \quad \ddot{\hat{\delta}} = \frac{\partial^2 \hat{\delta}}{\partial t^2} \quad (6c)$$

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



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

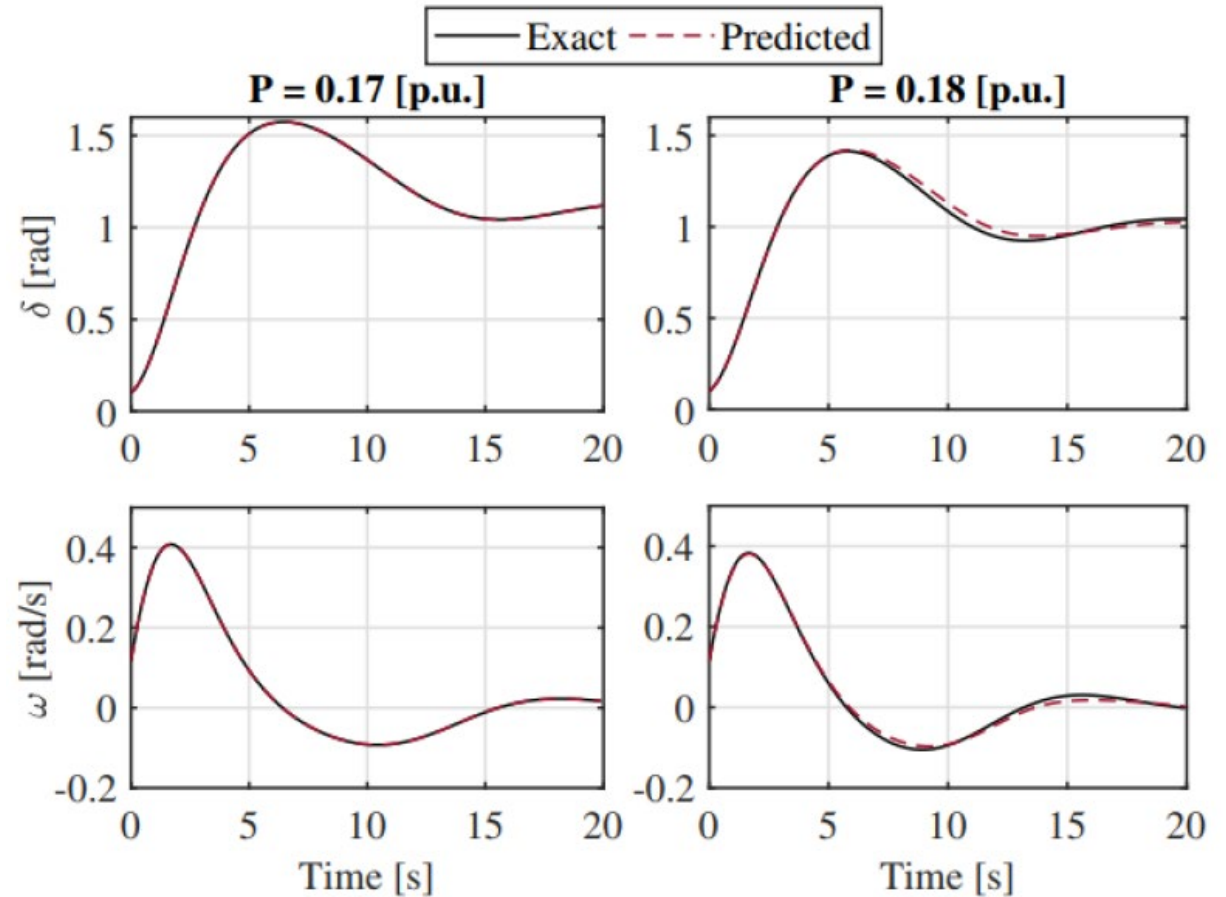
$$\min_{\mathbf{W}, \mathbf{b}} \quad \frac{1}{|N_\delta|} \sum_{i \in N_\delta} |\hat{\delta} - \delta^i|^2 + \boxed{\frac{1}{|N_f|} \sum_{i \in N_f} |f(\hat{\delta})|^2} \quad (6a)$$

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

$$\dot{\hat{\delta}} = \frac{\partial \hat{\delta}}{\partial t}, \quad \ddot{\hat{\delta}} = \frac{\partial^2 \hat{\delta}}{\partial t^2} \quad (6c)$$

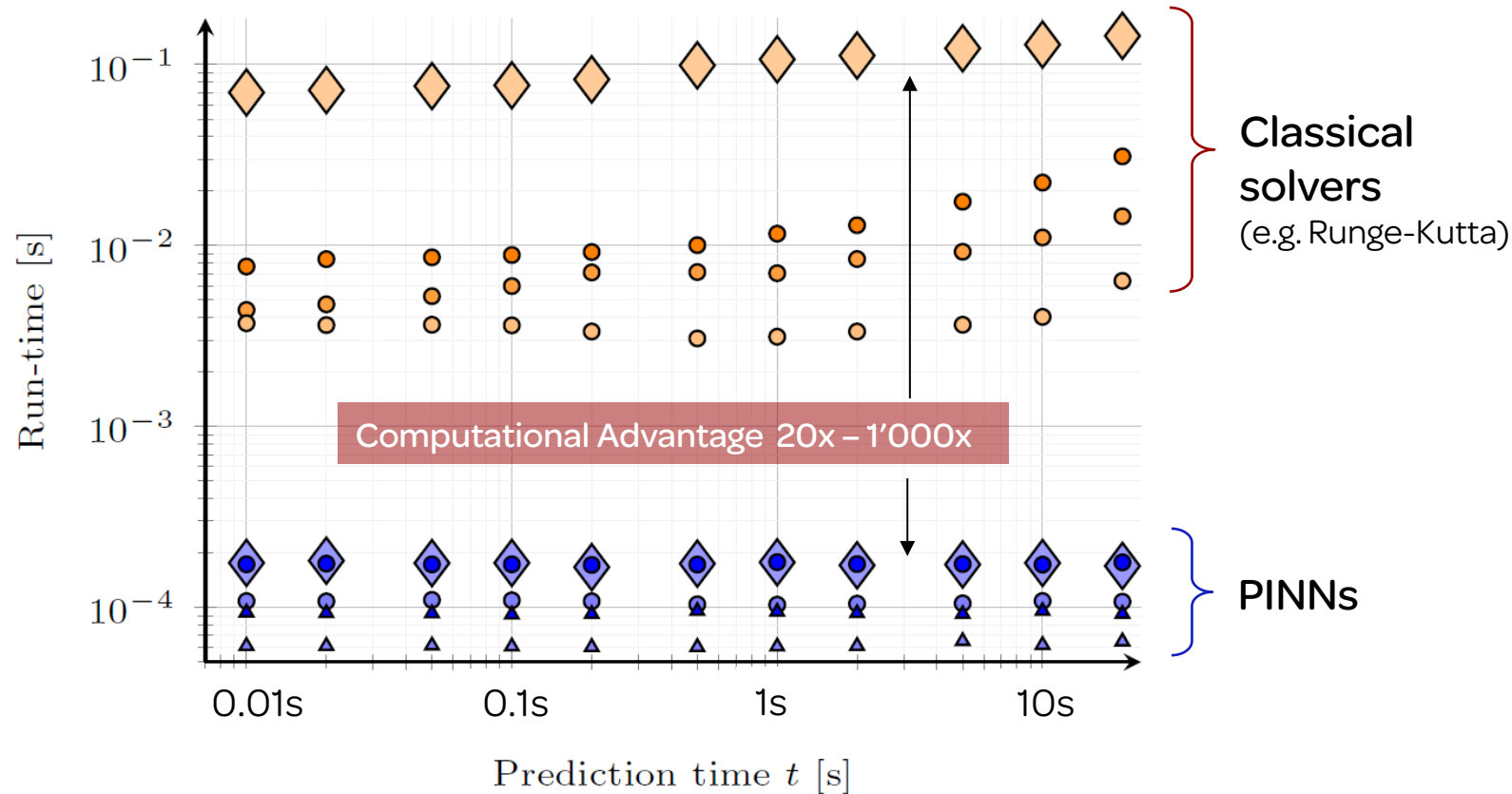
$$\boxed{f(\hat{\delta}) = M \ddot{\hat{\delta}} + D \dot{\hat{\delta}} + A \sin \hat{\delta} - P_m} \quad (6d)$$

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 NNs 100x faster than Classical numerical solvers



The further ahead we look in time, the larger the computational advantage is

Results from 11-bus and 39-bus

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

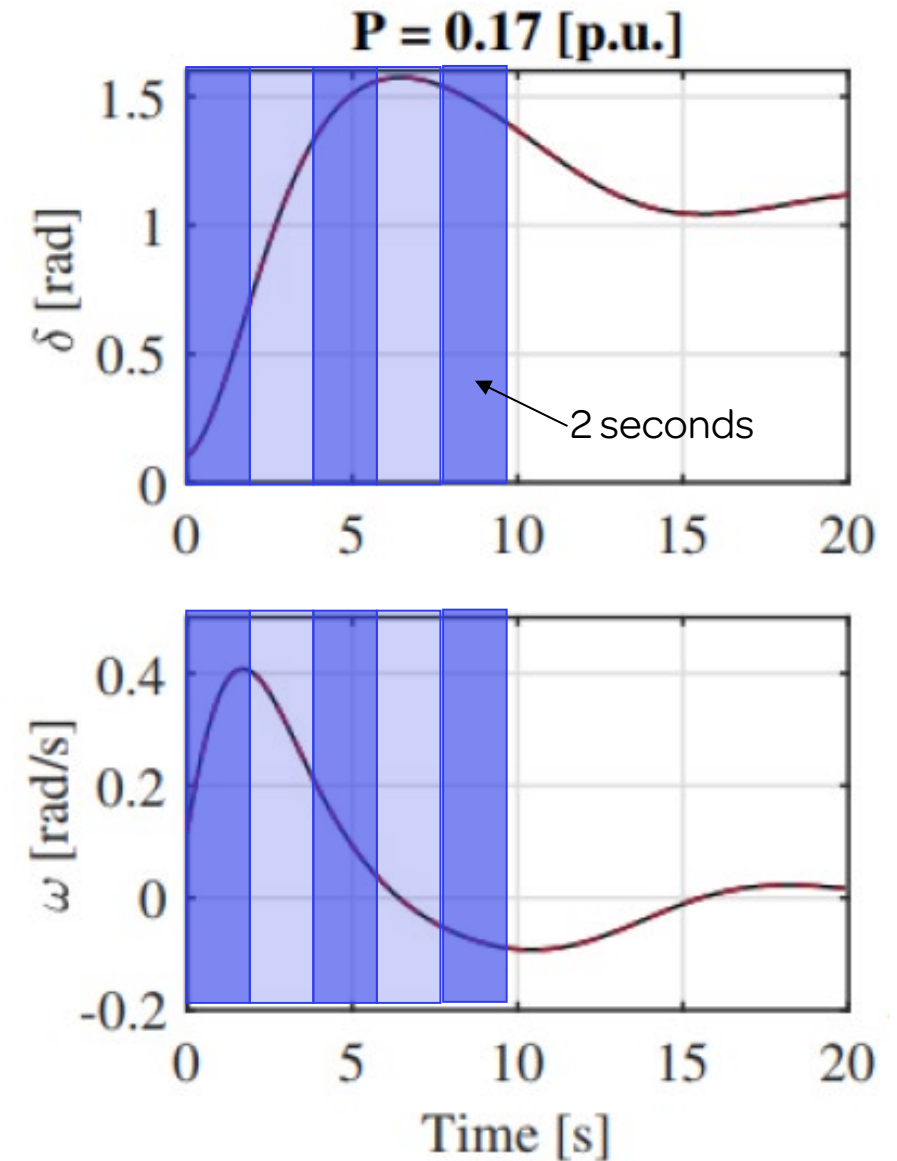
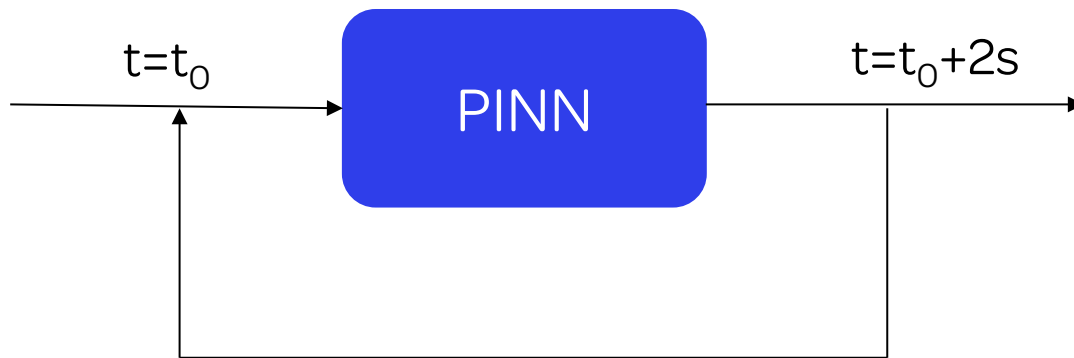
But, there is a trade-off. The further we look ahead in time, the more difficult the learning task becomes

Learning takes longer
PINN accuracy drops

What shall we do?

How can we reduce training time and improve performance?

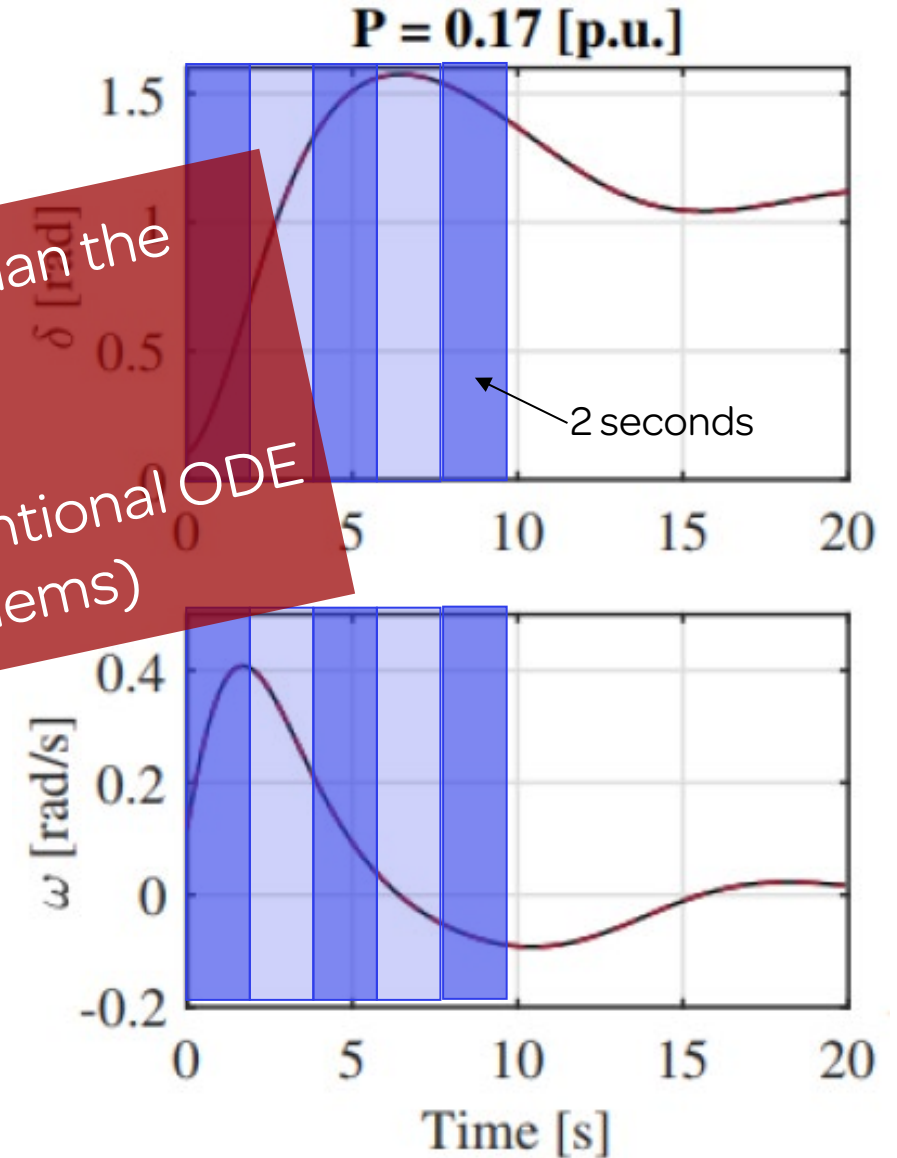
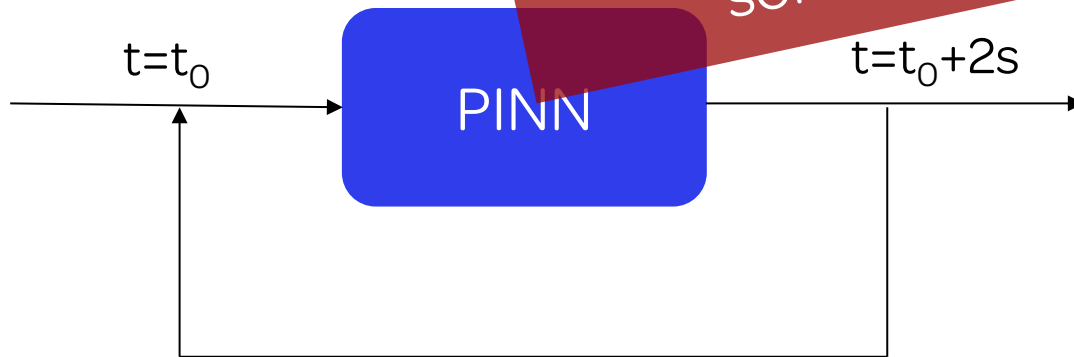
1. Train for a shorter time period but for a wide range of initial conditions
2. Use the PINN in a recurrent fashion

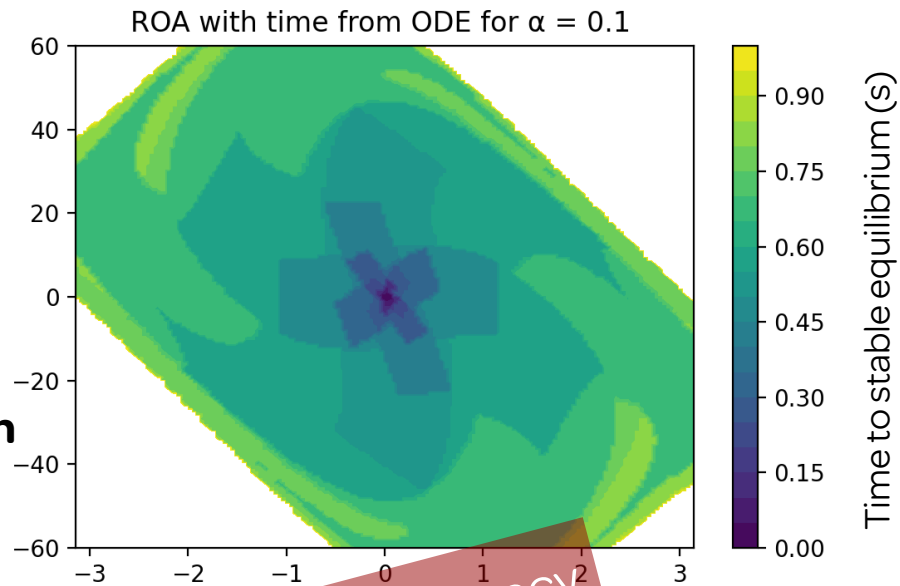


How can we reduce training time and improve performance?

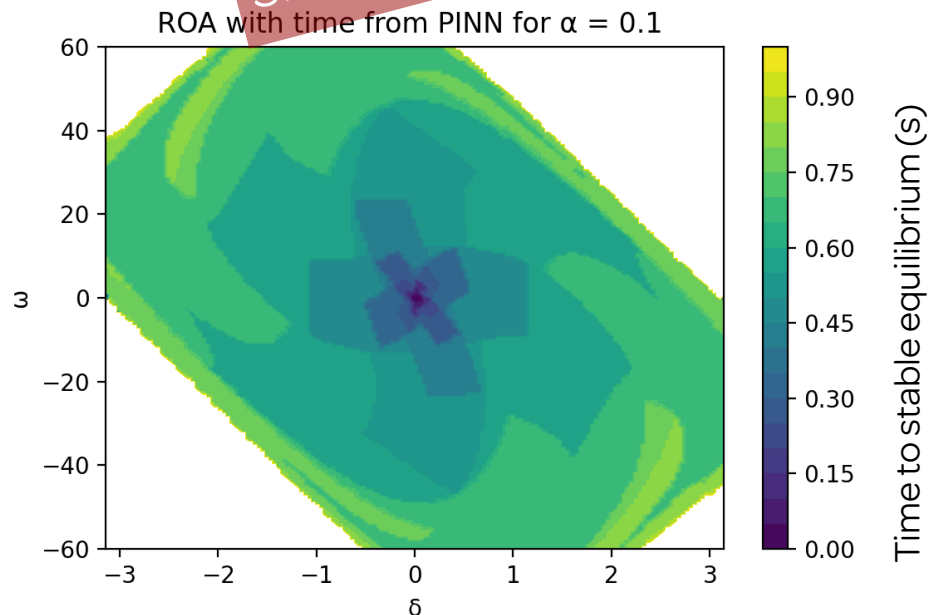
1. Train for a shorter time period but for a wide range of initial conditions
2. Use the PINN in a recurrent fashion

Is the recurrent PINN slower than the single-step PINN?
Yes.
But still 100x faster than conventional ODE solvers (for larger problems)



**Electro-
magnetic
Transient
(EMT)
Simulation**

Similar accuracy

PINN

Simulations for Wind Farms:

Estimating the Region of Attraction of a Wind Farm Controller

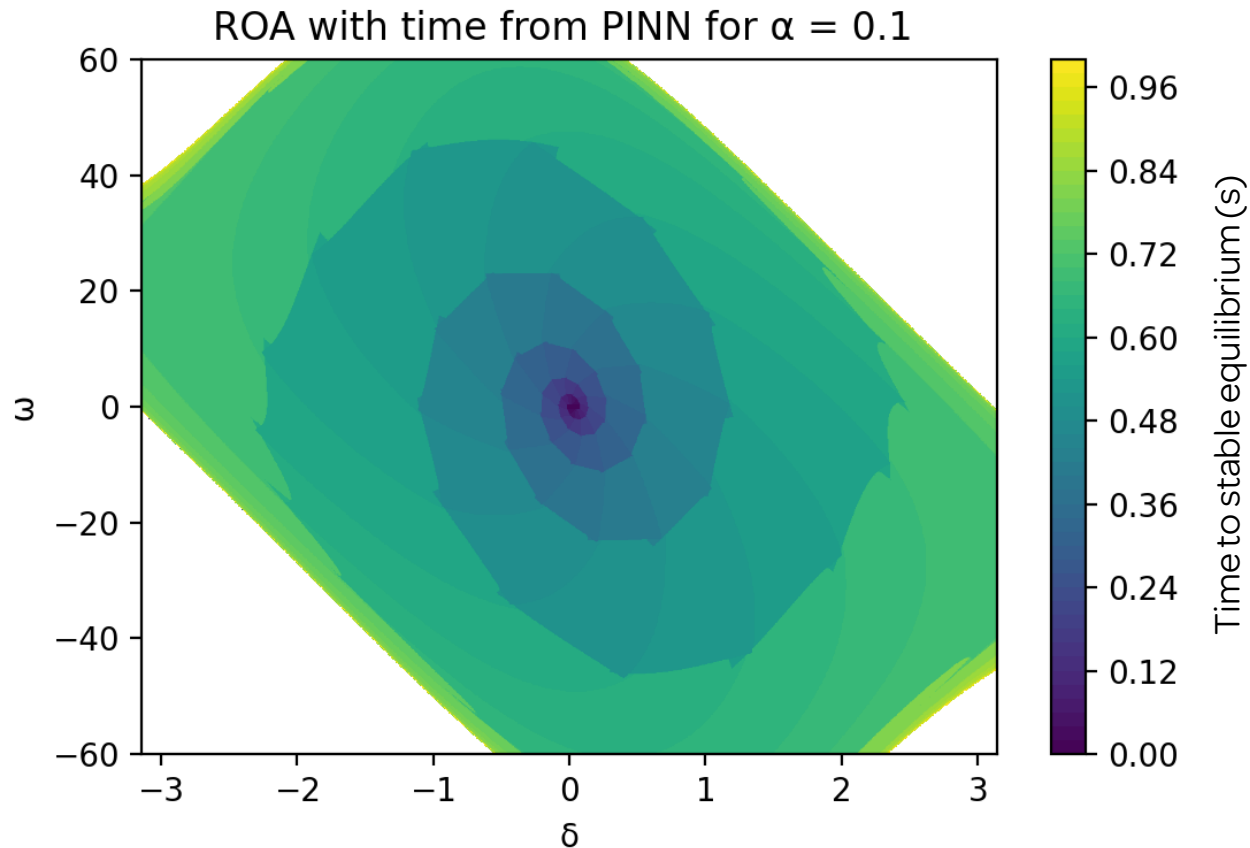
- Collaboration with Ørsted
 - Estimating the region of attraction of controllers is an important part of the wind farm design process
- Goal: Determine the best set of controller parameters (controller tuning)
- Training PINNs with GPUs

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



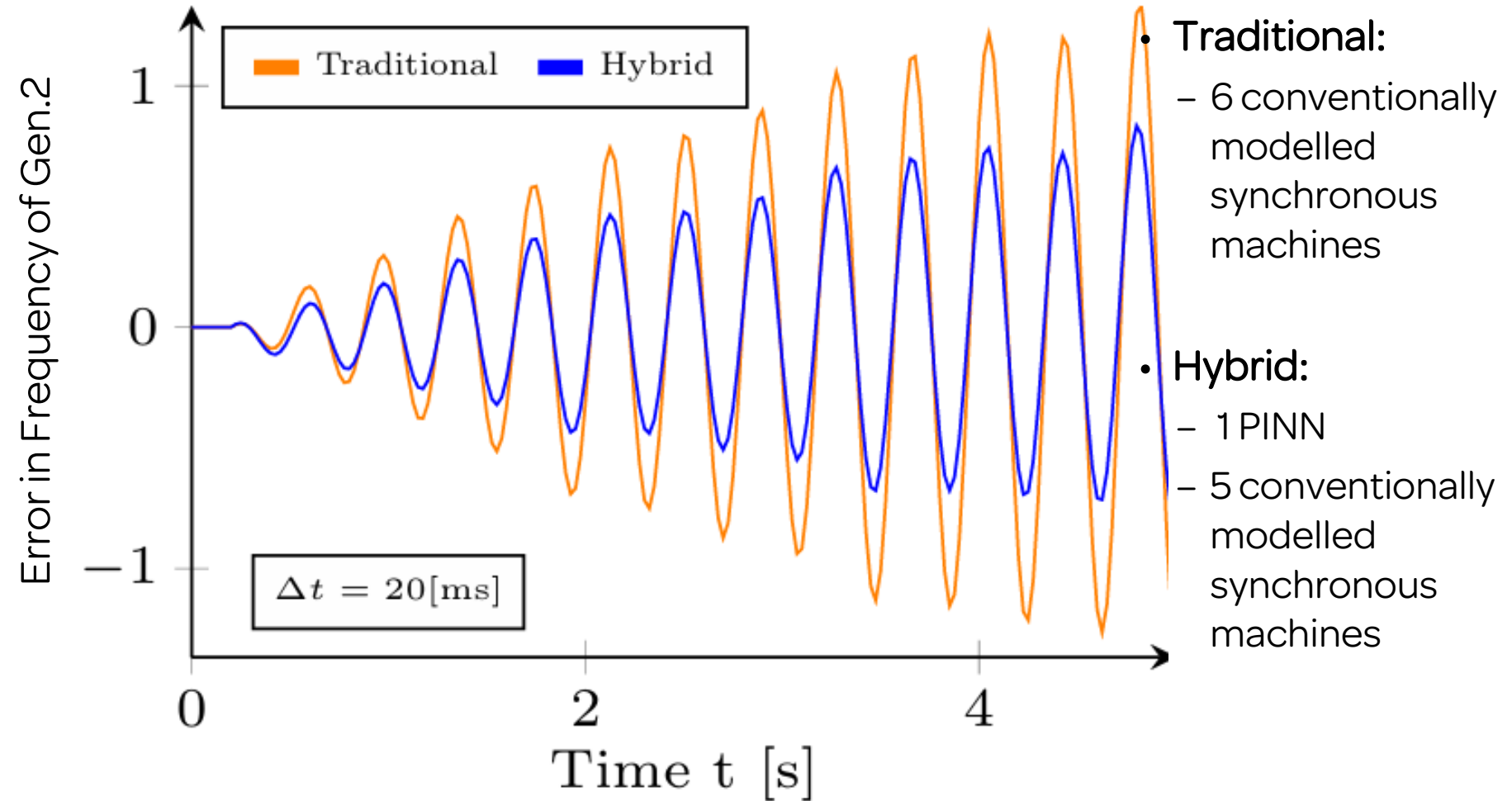
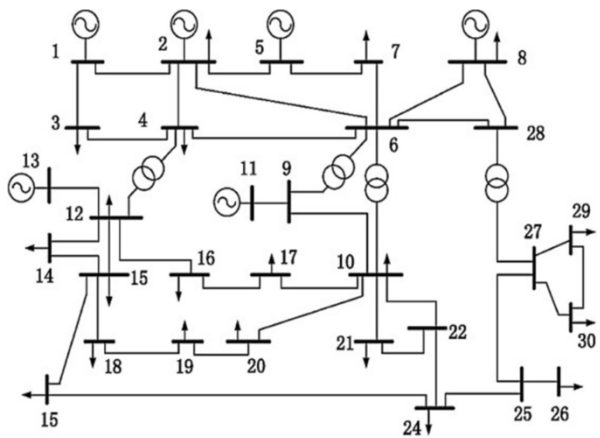
- Evaluation of 5 million points
 - EMT: ~2 days @ DTU HPC
 - PINNs: 90 minutes for training and 30 minutes for evaluation
- 25x – 100x faster
- Added benefit: once trained, PINN can run on a laptop

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 commercial simulators and accelerate them
2. Create a PINN-based Simulator → PINNSim

Integrating 1 PINN in IEEE 30-Bus System : Simulation error reduces (for the same time step)



Do you want to create your own PINNs?

Open-source Modular Python Toolbox!

[GitHub/radiakos/PowerPINN](https://github.com/radiakos/PowerPINN)

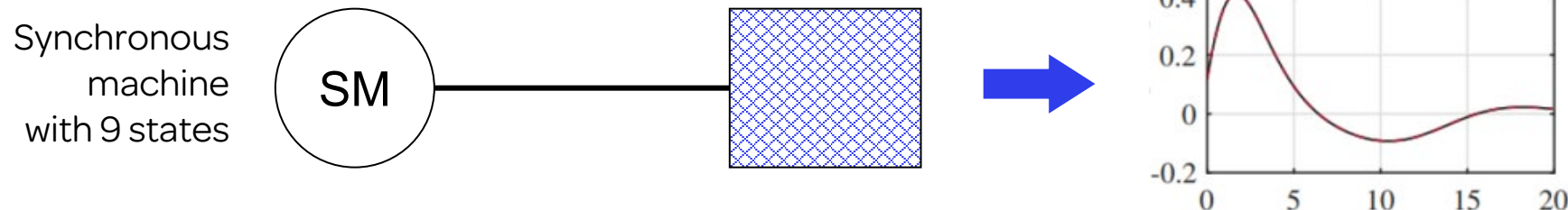


Components:

- 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



	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

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

Accelerate Power System Time-Domain Simulations in Commercial Tools

1. Create a tool that can assess which components will benefit by being replaced with a PINN (or, an ML model, in general)
2. Let the user decide if they want to “optimize” the simulator
3. Run the simulation with the “optimized” components, integrating PINNs in a plug-and-play fashion where necessary

First steps in that direction:

I. Ventura, J. Stiasny, S. Chatzivasileiadis, Physics-Informed Neural Networks: a Plug and Play Integration into Power System Dynamic Simulations. *Electric Power Systems Research*, vol. 248, 111885, 2025
<https://arxiv.org/abs/2404.13325>

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>

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

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

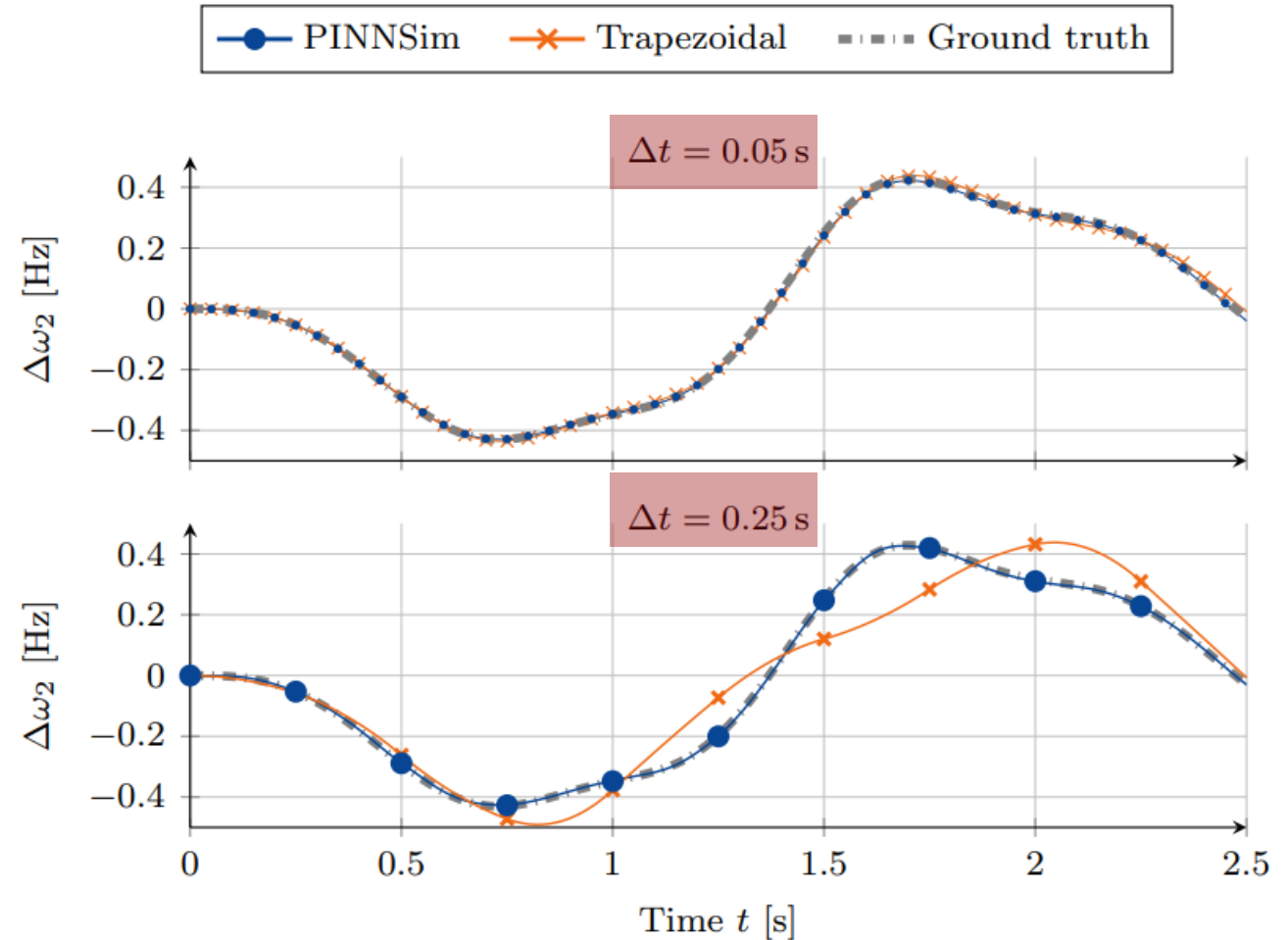
Short answer: probably not

What shall we do?

Physics-Informed Neural Networks for Power Systems: Vision (Part 2)

PINNSim: A modular power system time-domain simulator

- All components can be replaced by PINNs
- A library of component models implemented with Neural Networks
- “Drag’n’drop” to create your system
- Novelty: A new algorithm that integrates individual PINNs in a common simulation framework
- A completely new way of simulation which can be 10x-100x faster



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?

Physics-Informed Neural Networks for Power Systems: Vision (Part 3)

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 Eiras¹ Adel Bibi¹ Rudy Bunel² Krishnamurthy Dj Dvijotham² Philip H.S. Torr¹
M. Pawan Kumar²

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 Ellinas¹ Rahul Nellikath¹ Ignasi Ventura¹ Jochen Stiasny¹ Spyros Chatzivasileiadis¹

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 maximizes (1) indicates where the approximator has the

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

What would you do to make AI 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

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



European Research Council
Established by the European Commission

veriphied.ai



ai-effect.eu/

Safe AI by-design

Design AI
controllers
that
guarantee
asymptotic
stability

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](https://www.veriphied.ai)



ai-effect.eu/

Safe AI by-design

Design AI controllers that guarantee asymptotic stability

Use AI as Decision Support

1

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

2

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

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



European Research Council
Established by the European Commission

veriphied.ai



ai-effect.eu/

Safe AI by-design

Design AI controllers that guarantee asymptotic stability

Use AI as Decision Support

1

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

2

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

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
- **August 2024:** AI Act is an official EU Regulation



A lot of recent developments for trustworthy AI

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

Gurobi Optimizer

Gurobi 10.0 also includes the following advances in the underlying algorithmic framework:

- ✓ **New network simplex algorithm** – Greatly speeds up solving LPs with network structure.
- ✓ **New heuristic for QUBO models, which can arise in quantum optimization** – Improves Gurobi's ability to quickly find good feasible solutions for quadratic unconstrained Boolean optimization problems.
- ✓ **Significant performance gains on MIPs that contain machine learning models** – Results in a more than 10x improvement on certain models that contain embedded neural networks with ReLU activation functions.

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
- **Recent focus on AC-Optimal Power Flow (NLP): 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-Optimal Power Flow

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

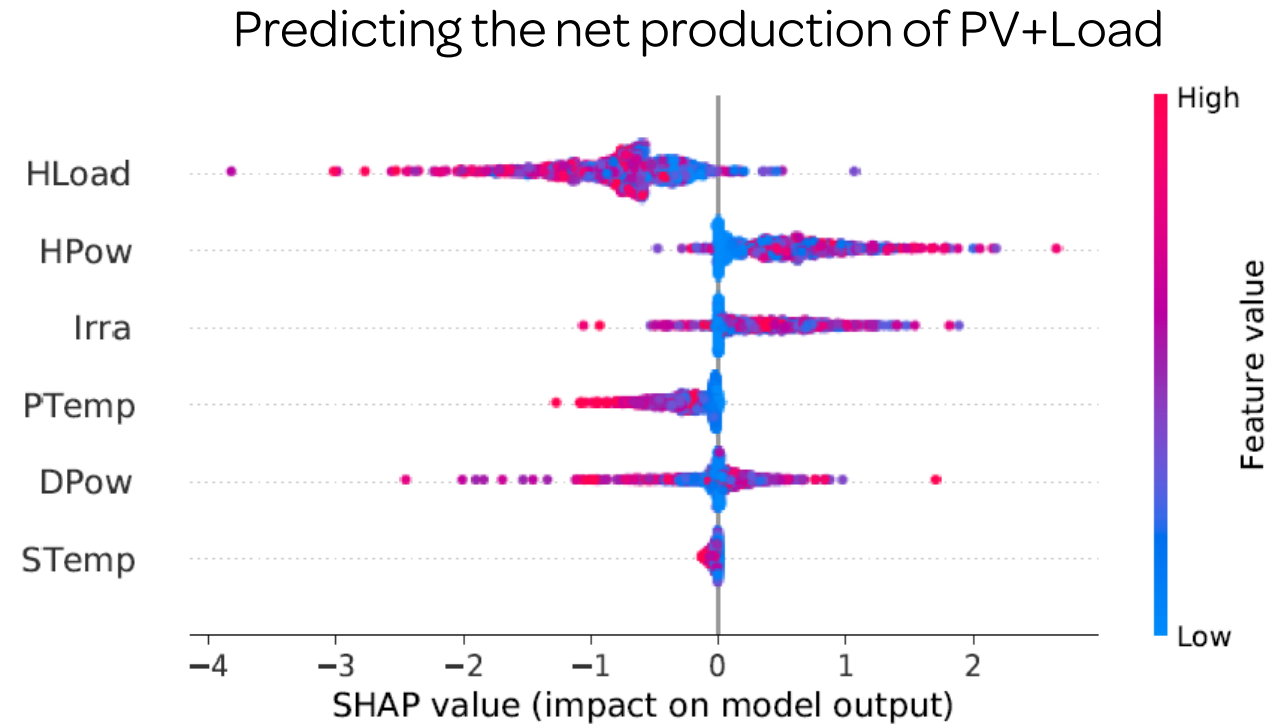
S. Chevalier, I. Murzakhonov, S. Chatzivasileiadis, *GPU-Accelerated Verification of Machine Learning Models for Power Systems*, **Best Paper Award at HICSS** (Hawaii International Conferences on Systems Sciences), Jan. 2024 <https://arxiv.org/pdf/2306.10617>

A lot of recent developments for trustworthy AI

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

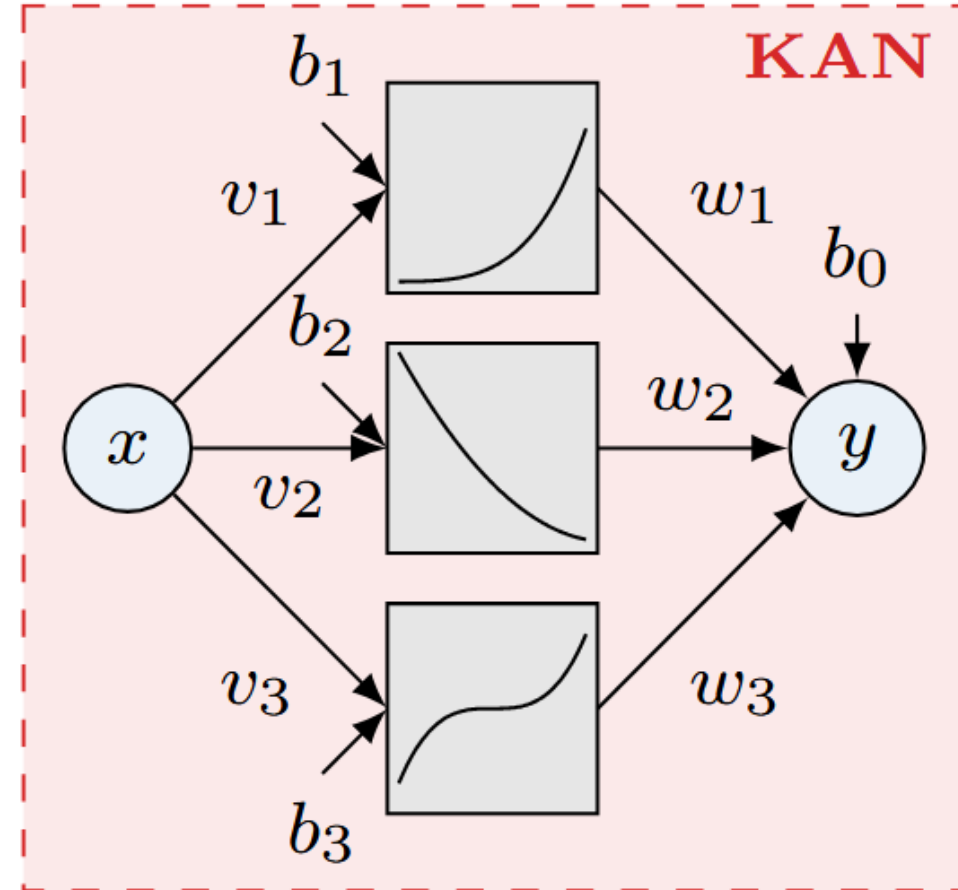
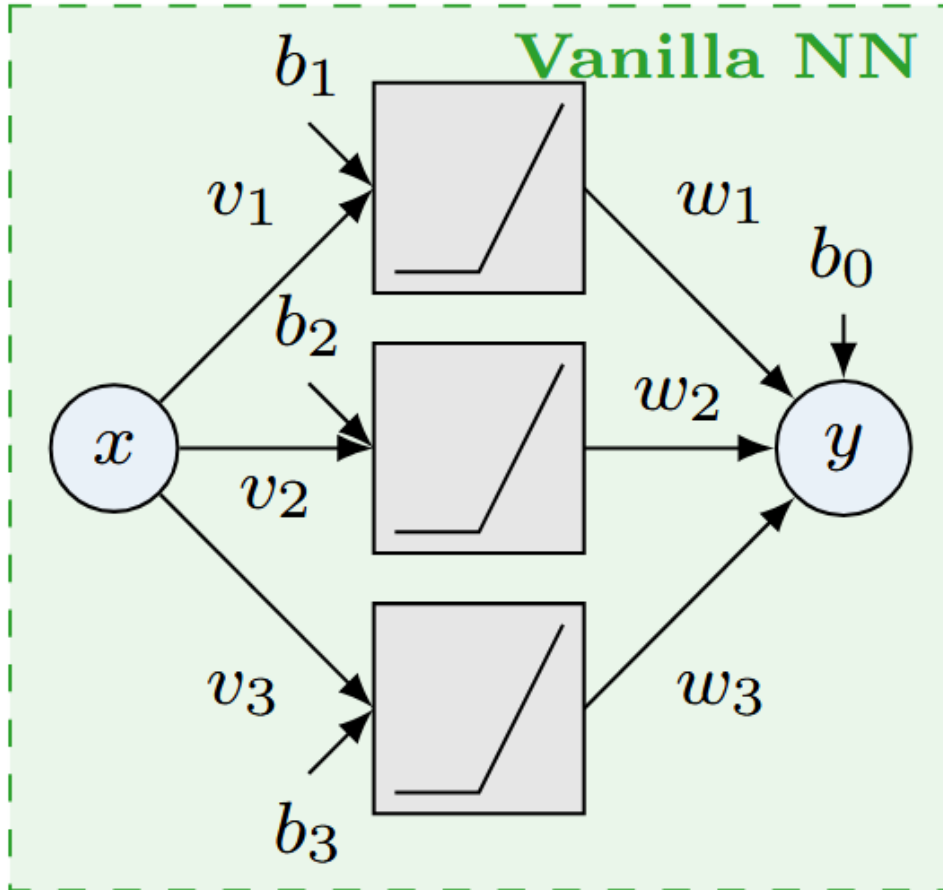


<https://shap.readthedocs.io/en/latest/>



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](#)]

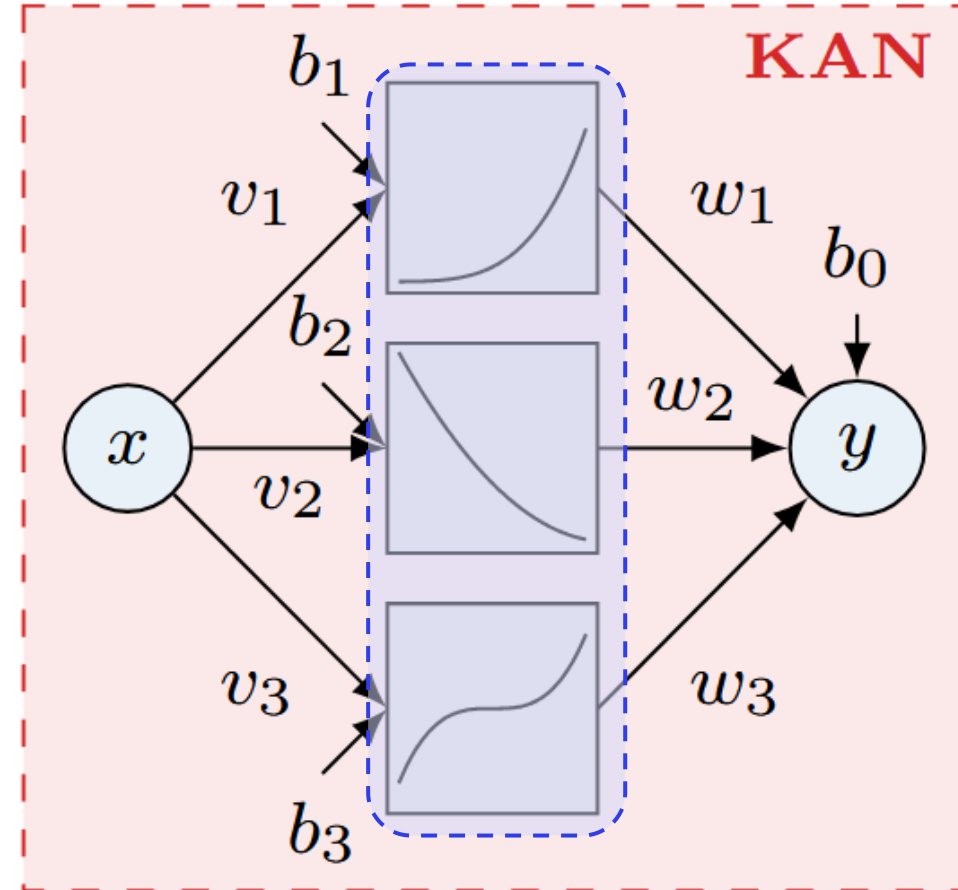
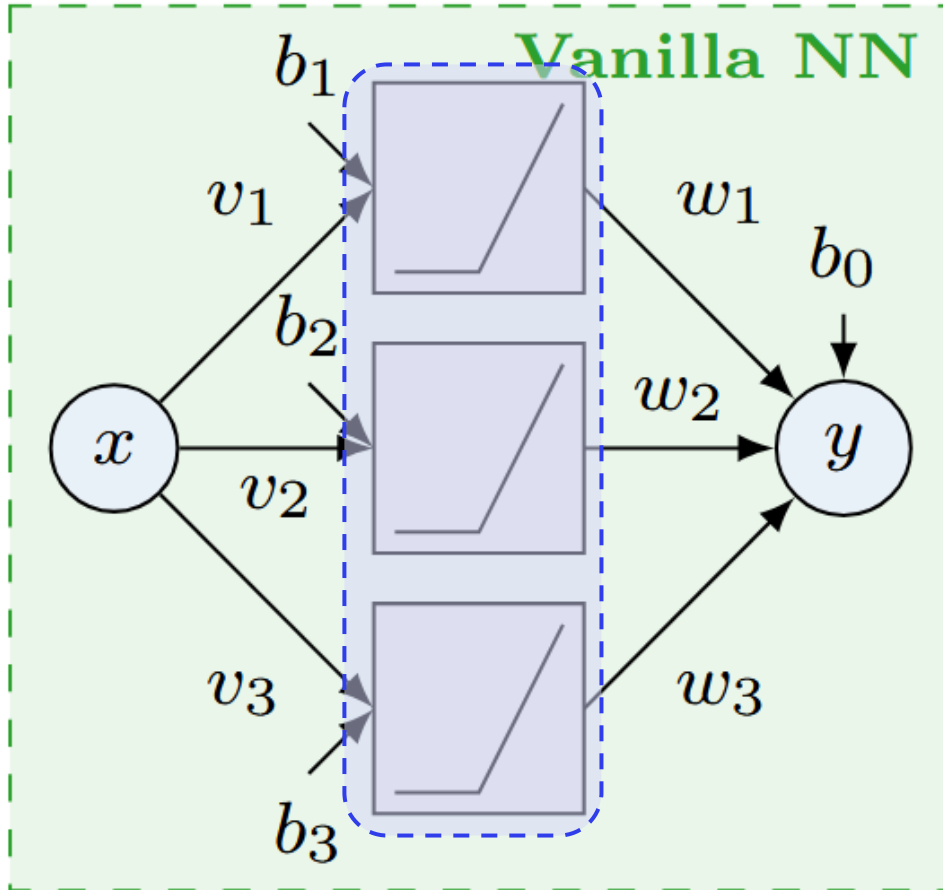
Kolmogorov Arnold Networks



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>

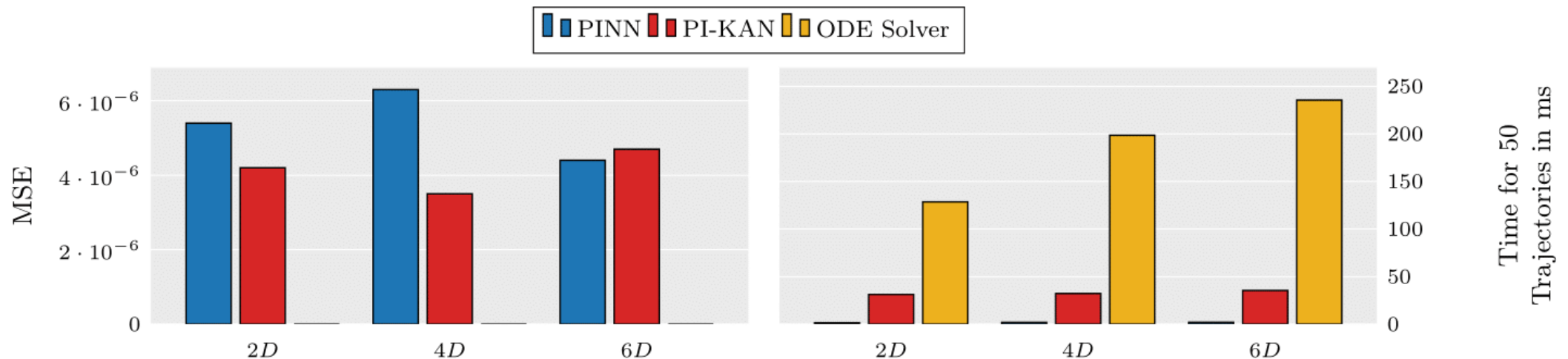
P. Ellinas, I. Karampinis, I. Ventura-Nadal, R. Nellikkath, J. Vorwerk, S. Chatzivasileiadis, Physics-Informed Machine Learning for Power System Dynamics: A Framework Incorporating Trustworthiness, *Sustainable Energy, Grids and Networks*, Elsevier, 2025. <https://doi.org/10.1016/j.segan.2025.101818>

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>

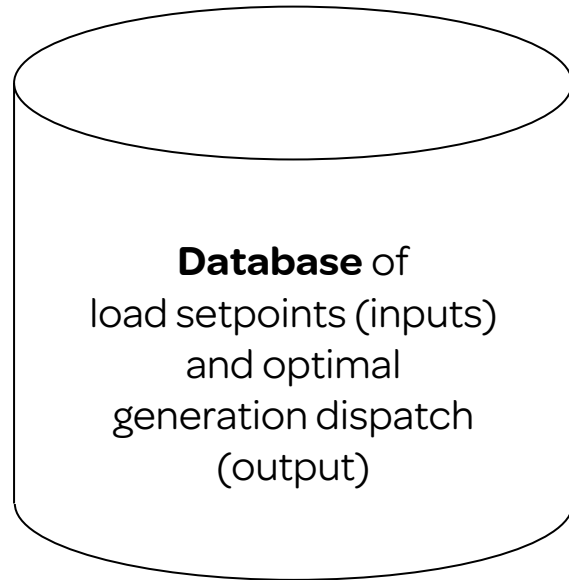
Neural Network Verification and Provable Worst-Case Guarantees for Power Systems

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>

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>

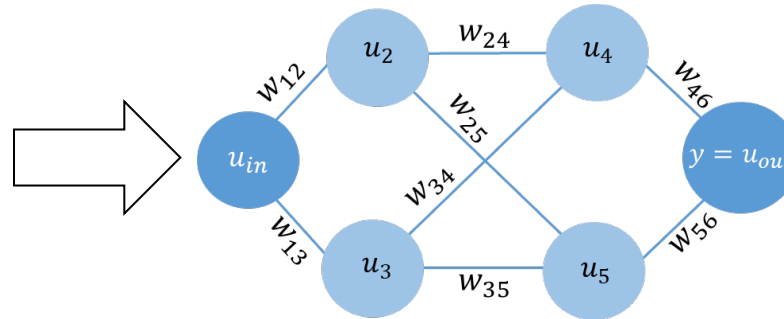
V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in International Conference on Learning Representations (ICLR 2019), 2019

Guiding Application: Optimal Power Flow with Neural Networks (can apply to any optimal control problem)



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

Approaches proposed up to now

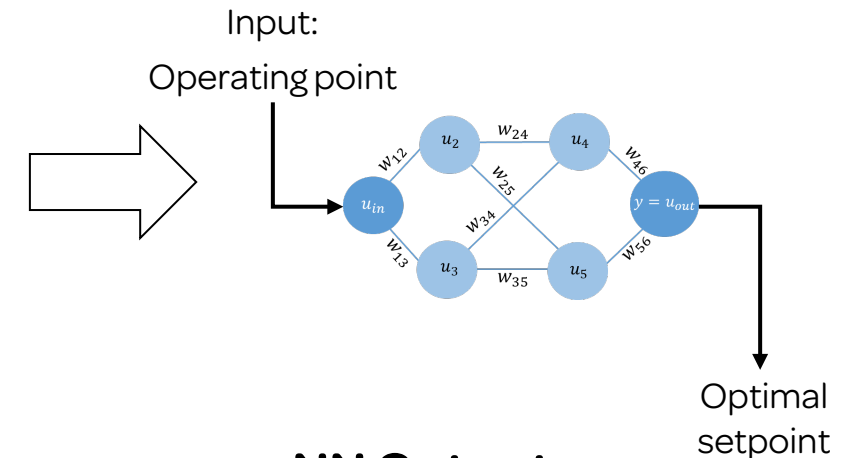


2. Train a neural network

3. Test the neural network

4. Is accuracy high enough?

5. Use the NN



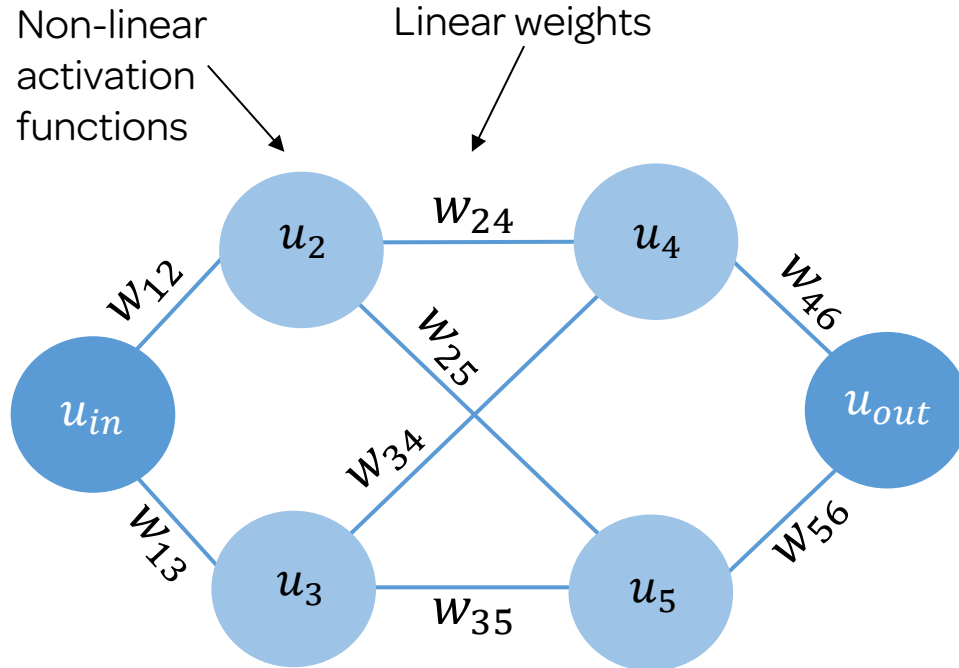
NN Output:

Optimal generator
dispatch

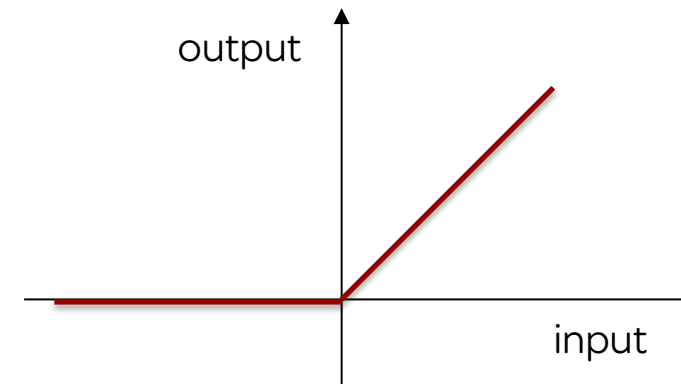
Extremely fast: up to
100x-1'000x faster

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

From Neural Networks to Mixed-Integer Linear Programming

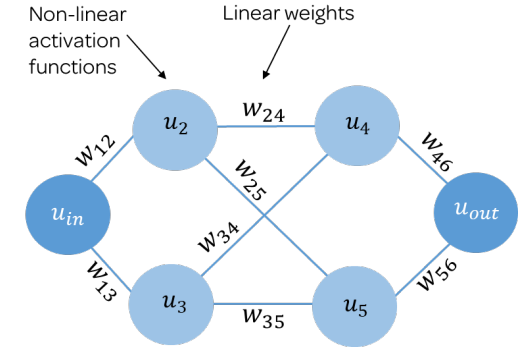
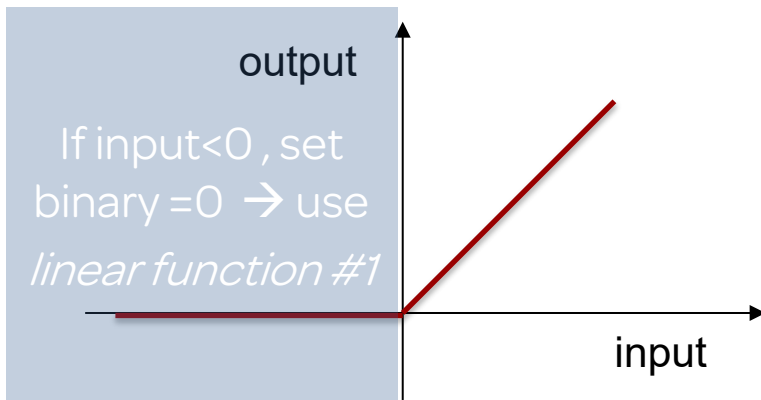


- Most usual activation function: ReLU
- **ReLU**: Rectifier Linear Unit

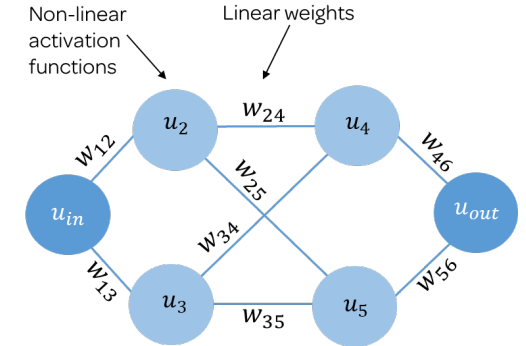


From Neural Networks to Mixed-Integer Linear Programming

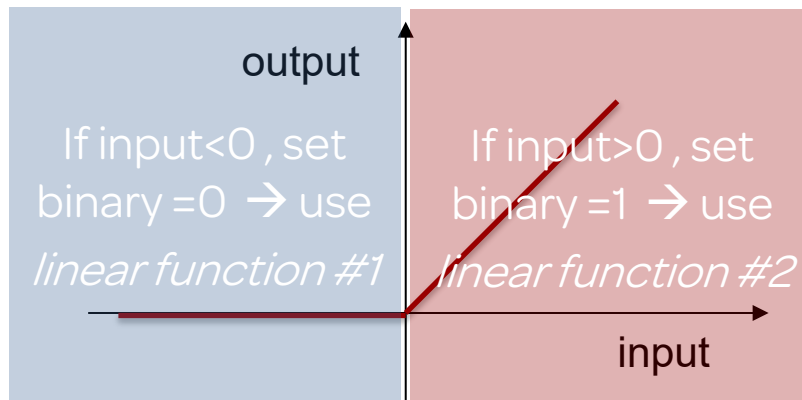
1. But **ReLU** can be transformed to a **piecewise linear function with binary variables**



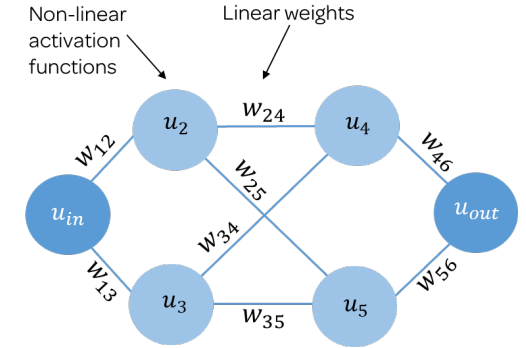
From Neural Networks to Mixed-Integer Linear Programming



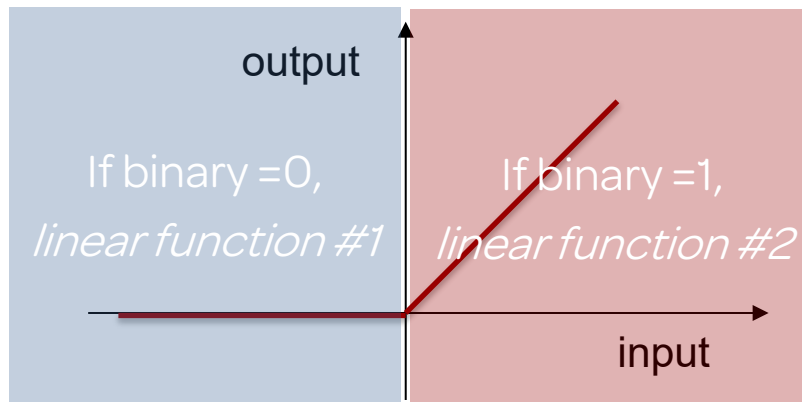
1. But **ReLU** can be transformed to a **piecewise linear function with binary variables**



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**

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)

Verify Maximum limit-violations

1. Maximum violation of generator limits

$$\nu_g = \max(\hat{\mathbf{p}}_g - \mathbf{p}_g^{\max}, \mathbf{p}_g^{\min} - \hat{\mathbf{p}}_g, \mathbf{0})$$

$$\begin{array}{ll} \max & \nu_g \\ \text{s.t.} & \mathbf{A}_d \mathbf{p}_d \leq \mathbf{b}_d \quad \text{Convex polytope as input domain } \mathcal{D} \\ & \hat{\mathbf{p}}_g = NN(\mathbf{p}_d) \quad \text{Mixed-integer reformulation of trained NN} \end{array}$$

Example:

$$0.6 \mathbf{p}_d^{\max} \leq \mathbf{p}_d \leq 1.0 \mathbf{p}_d^{\max}$$

Electric load setpoints vary between 60% and 100% of their rated value

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)
<i>case9</i>				
<i>case30</i>				
<i>case39</i>				
<i>case57</i>				
<i>case118</i>				
<i>case162</i>				
<i>case300</i>				

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)
<i>case9</i>	2.5	1.8	2.8	1.9
<i>case30</i>	1.7	0.6	3.6	3.1
<i>case39</i>	51.9	37.2	270.6	120.0
<i>case57</i>	4.2	0.0	23.7	0.0
<i>case118</i>	149.4	15.6	997.8	510.8
<i>case162</i>	228.0	180.0	1563.3	974.1
<i>case300</i>	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**

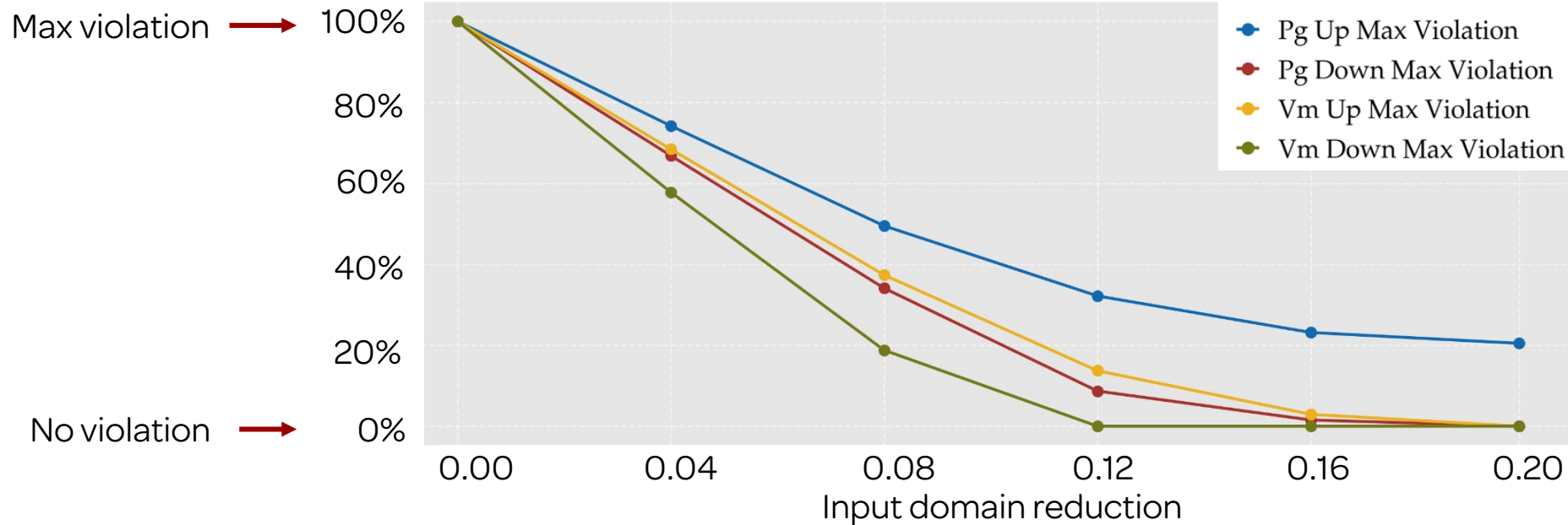
	Empirical lower bound		Exact worst-case guarantee	
Test cases	ν_g (MW)	ν_{line} (MW)	ν_g (MW)	ν_{line} (MW)
<i>case9</i>	2.5	1.8	2.8	1.9
<i>case30</i>	1.7	0.6	3.6	3.1
<i>case39</i>	51.9	37.2	270.6	120.0
<i>case57</i>	4.2	0.0	23.7	0.0
<i>case118</i>	149.4	15.6	997.8	510.8
<i>case162</i>	228.0	180.0	1563.3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

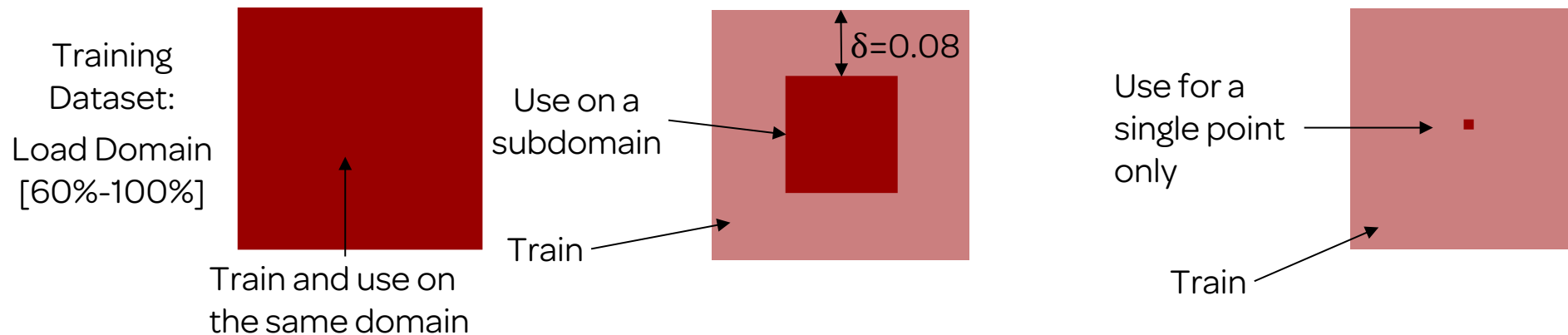
Reducing the worst-case violations: Train for a larger domain; deploy on a subdomain



118-bus system

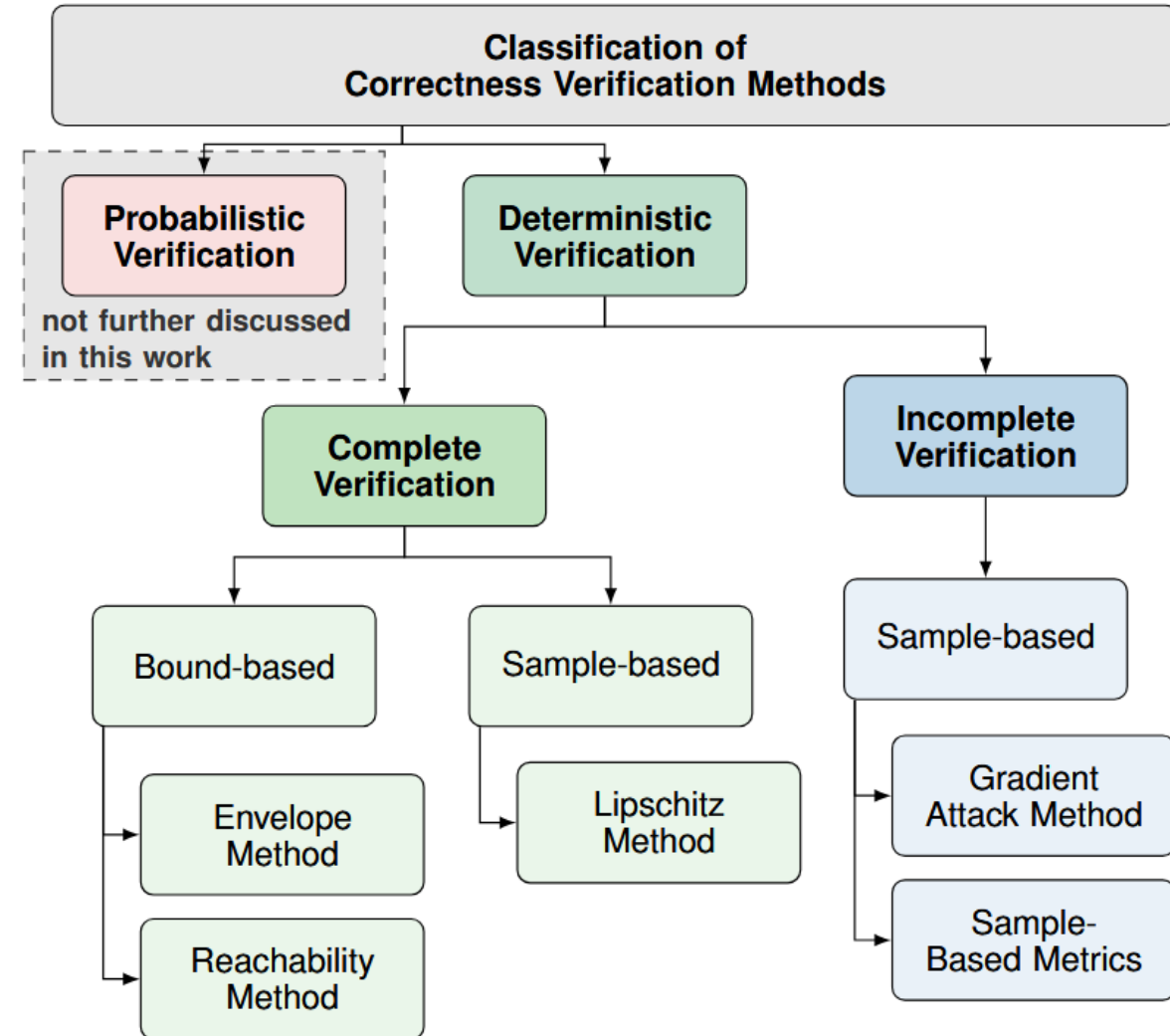
Non-Linear
Optimization
Problem
(AC-OPF)

Worst-case
violations appear to
be at the boundary
of the input domain



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 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%."



AI-EFFECT EU project

Start: 1st October 2024

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

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 training processes, however, are designed to deliver ML with good *average* performance, but do not offer any guarantees about their worst-case estimation error. For safety-critical systems such as power systems, this places a major barrier to adoption. So far, approaches could determine the worst-case violations of only *trained* ML algorithms. To the best of our knowledge, this is the first paper to introduce a neural network training procedure designed to achieve both a good average performance and minimum worst-case violations. Using the AC Power Flow (OPF) problem as a guiding application, our

fast surrogate functions in place of intractable combi-level optimization problems to make them computationally feasible [11]. These developments have led researchers to focus on the development of advanced ML architectures, especially neural networks (NN), with improved accuracy for power system applications. One of the recent developments among them is, for example, Informed Neural Networks (PINNs) which incorporate physical equations governing the power flow into the training process [12]–[17]. PINNs can achieve higher prediction accuracy

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

Some Final Thoughts

- Physics-Informed Machine Learning has the potential to accelerate computations by 10x-100x
- Integrate Physics-Informed Neural Networks in commercial simulators
- PINNSim: Create completely new simulation tools based on PINNs
 - Do not need a single NN for the whole problem
 - **Let's work with "Libraries of Neural Networks",** similar to "Libraries of Models"

Some Final Thoughts

- Physics-Informed Machine Learning has the potential to accelerate computations by 10x-100x
- Integrate Physics-Informed Neural Networks in commercial simulators
- PINNSim: Create completely new simulation tools based on PINNs
 - Do not need a single NN for the whole problem
 - **Let's work with "Libraries of Neural Networks",** similar to "Libraries of Models"
- **Trustworthy AI is necessary** if we are to deploy it in safety-critical systems
- Neural Network Verification and Provable Worst-Case Guarantees:
 - can benchmark AI tools
 - (similar to what convex relaxations have done for non-convex optimization programs)
- Need for Testing Facilities that validate AI tools → towards AI standards

Still, a lot of challenges our community needs to address:

- How can PINNs efficiently capture stiff dynamics?
- How can we scale NN verification for non-linear problems?
- ... and many more

Open-source Toolboxes

1. Generate your own training datasets!

[GitHub/bastiengiraud/DSA-learn](https://github.com/bastiengiraud/DSA-learn)

2. Train your own Physics-Informed Neural Networks!

[GitHub/radiakos/PowerPINN](https://github.com/radiakos/PowerPINN)

3. Play with a PINN for Power System Dynamics!

[Google Colab PINN Playground](https://colab.research.google.com/github/radiakos/PowerPINN/blob/main/colab/PINN_Playground.ipynb)



You can find them all in:

www.chatziva.com/downloads.html

Thank you!



Spyros Chatzivasileiadis

Professor

www.chatziva.com

spchatz@dtu.dk

