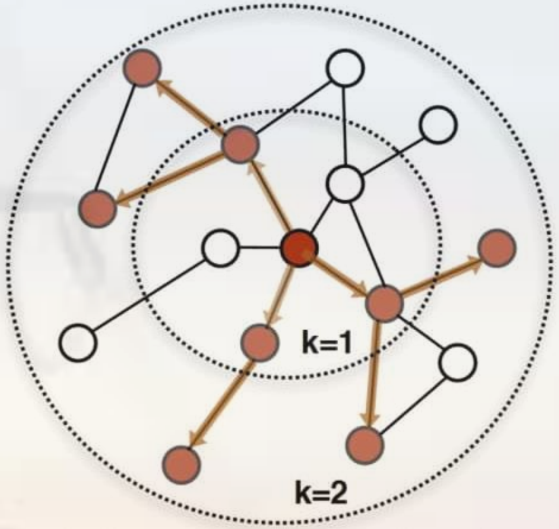
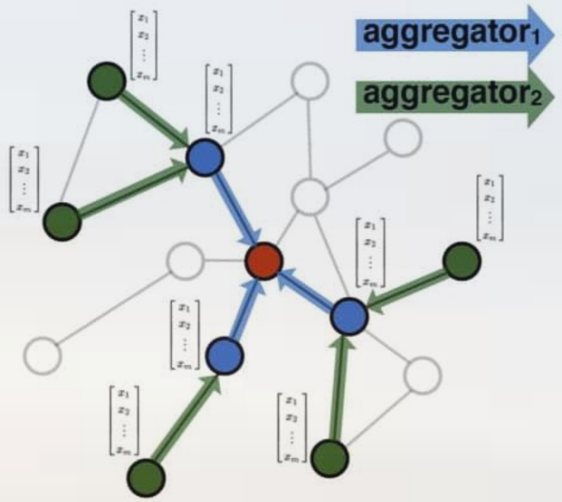




Opportunities & challenges in graph-based learning for power system application

TU Delft

Jochen Cremer

Delft AI Energy Lab

Mission & objective

- combine groundbreaking ML with the reliable theory of the physical energy system
- make energy systems sustainable, reliable, effective

Education

- EE4C12 ML for Electrical Engineering
- SET 3125 Machine Learning Workflows for Digital Energy Systems
- SC42150 Statistical Signal Processing
- SC42110 Dynamic Programming and Stochastic Control
- MOOC Digitalization of Intelligent and Integrated Energy Systems
- Crash course of “Data-science”

Research

- Supervised learning for real-time grid assessment
- Distributed learning for power system congestion management
- Data-driven grid models for electricity load and weather forecasts
- Characterizing healthy/normal trajectories of complex dynamical systems using dictionary learning
- From fast Fourier transform to fast reinforcement learning

Key innovations

- AI-based algorithms for grid operation
- Real-time security assessment and anomaly detection
- Real-time learning algorithms for control and security of complex dynamical systems

alliander | AIT AUSTRIAN INSTITUTE OF TECHNOLOGY | TenneT | STEDIN.NET

TU Delft | AI Initiative

https://www.tudelft.nl/ai/delft-ai-energy-lab

DAIEnergyLab

DAI-Energy-Lab@tudelft.nl

Team


Jochen Cremer


Peyman Mohajerin Esfahani


Haiwei Xie


Olayiwola Arowolo


Shabnam Khodakaramzadeh


Mert Karaçelebi


Basel Morsy


Mohammad Boveiri


Ali Rajaei


Jochen Stiasny


Betuel Mamudi


Viktor Zobernig


Demetris Chrysostomou


Benjamin Habib


Luca Hofstadler


Runyao Yu


Perine Cunat

Acknowledgements


Benjamin Habib


Elvin Isufi


Ward van Breda

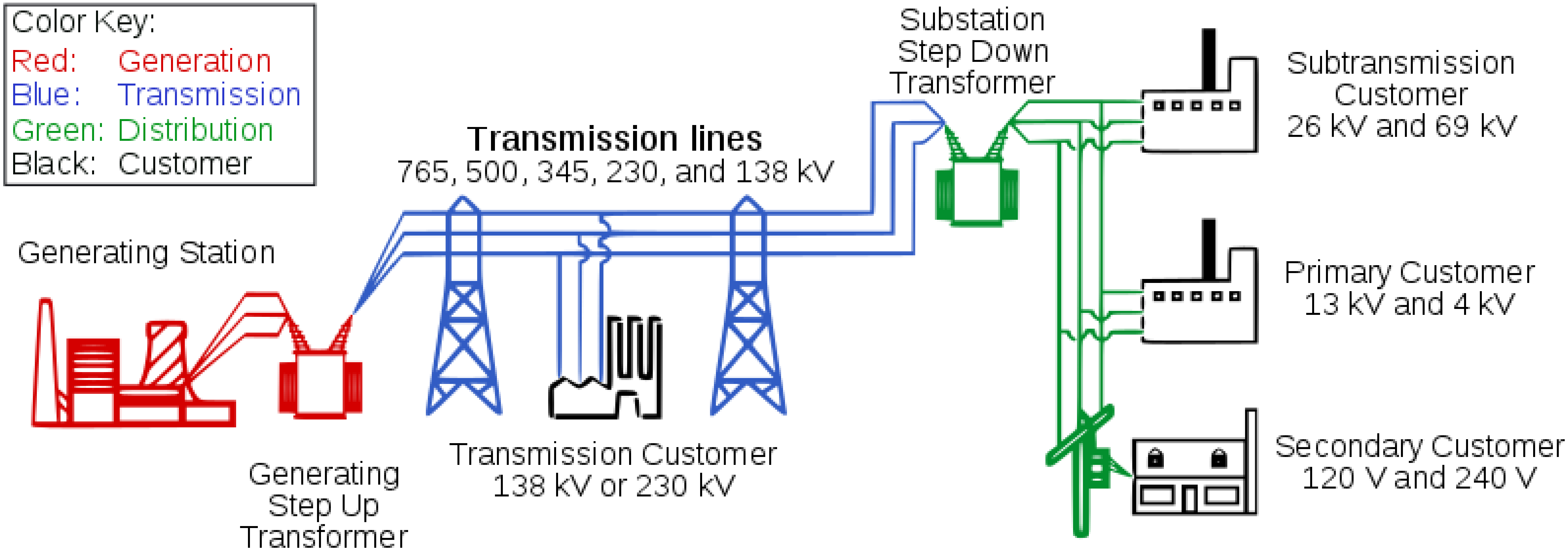

Arjen Jongepier


Olayiwolo Arowolo

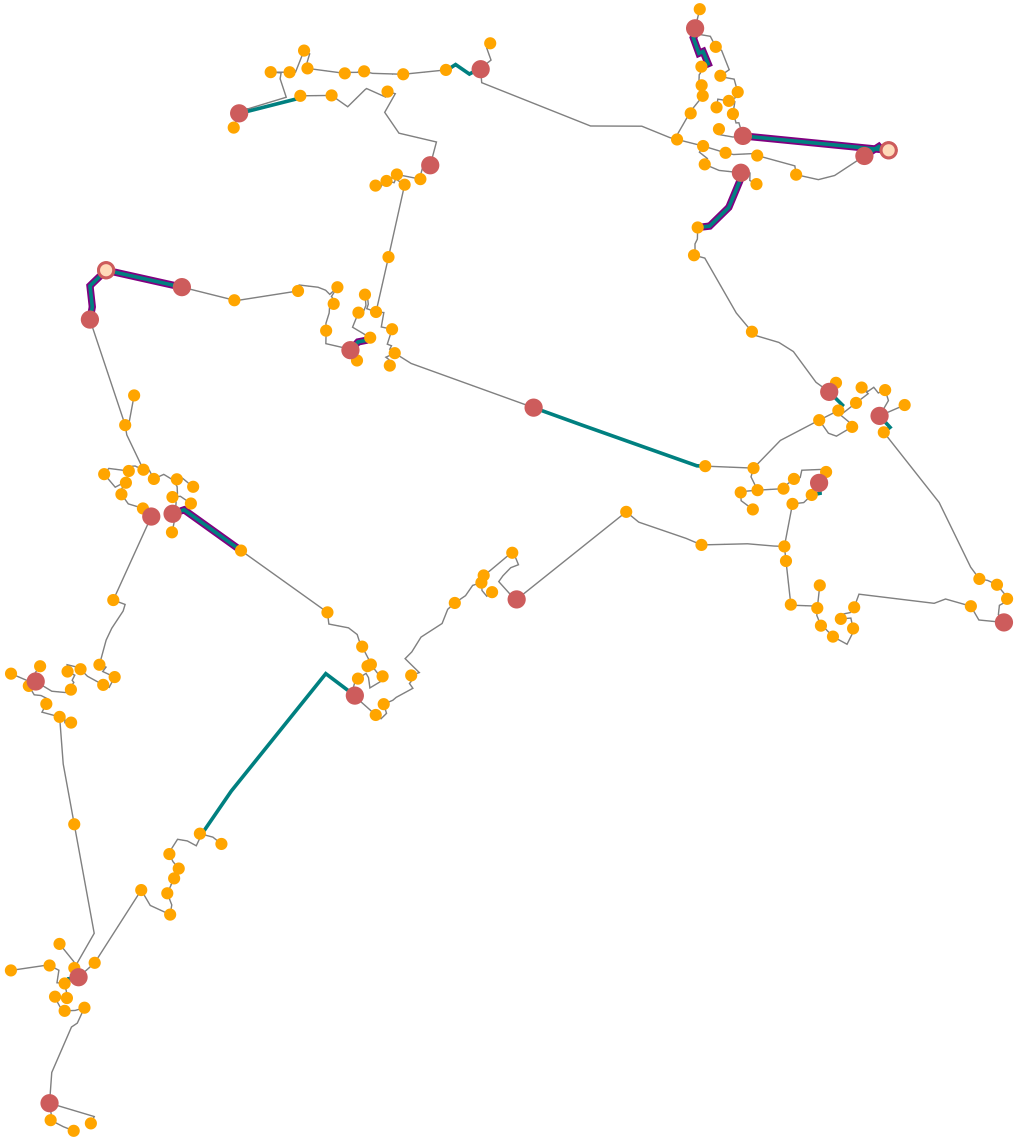

Jochen Stiasny


Betuel Mamudi

Power systems



Distribution system (MV)

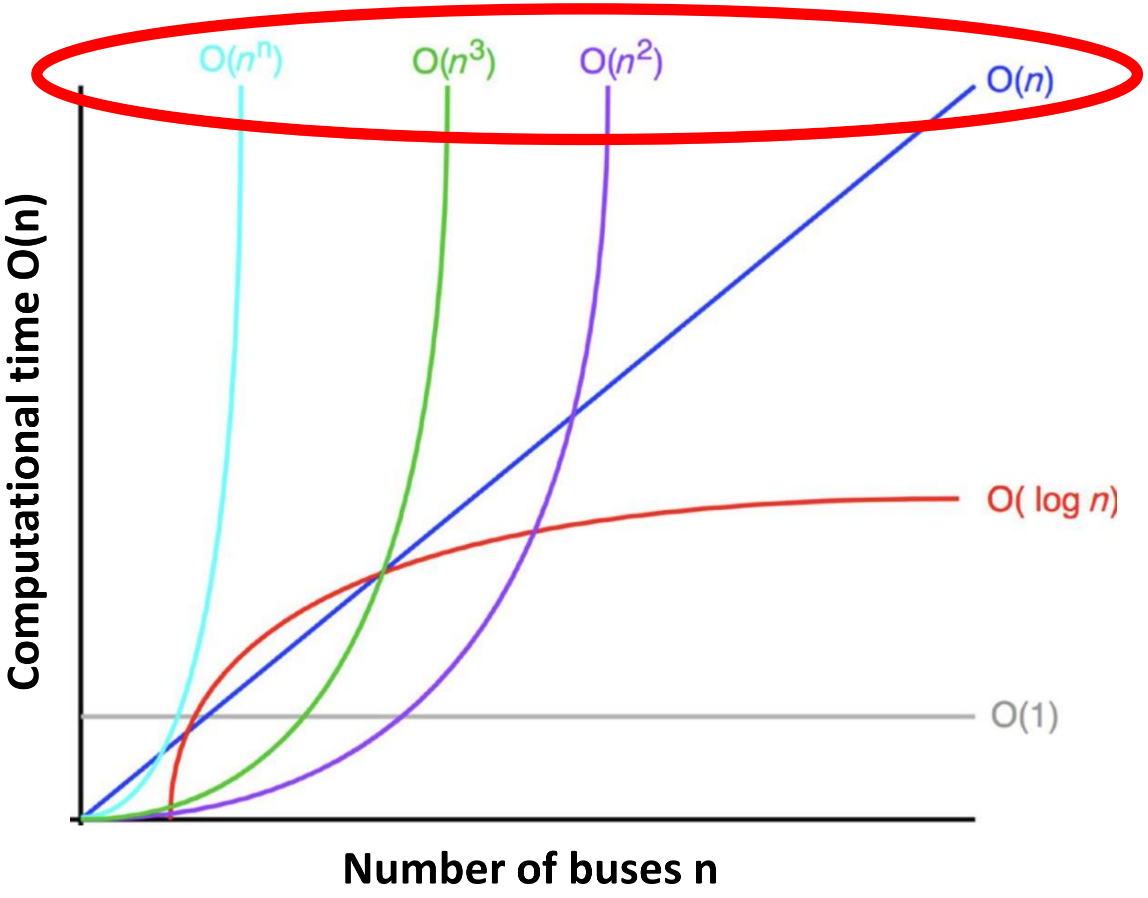


- Trafo
- Lines
- MV/LV buses
- HV buses
- Power flow measurement
- Voltage measurements
- Focus bus

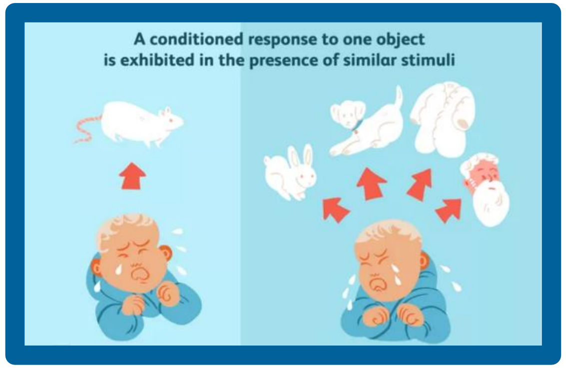
TU Delft

Scaling power system analysis to very large grids

Example: reliability assessment

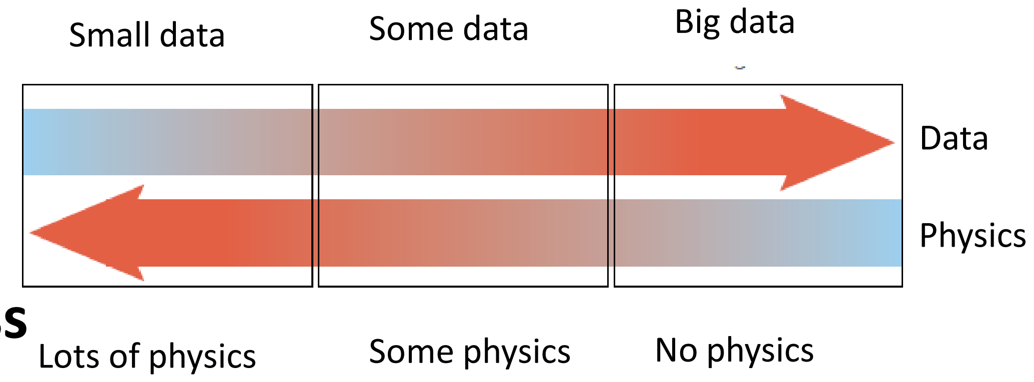


What are barriers applying AI?



1. **Generalising to out-of-distribution data**
   - Different energy scenarios in the future
   - Historical observations are not suitable
   - Energy systems will be ever-evolving

2. **Large training data is needed**
   - Power systems are extremely large systems
   - Training an AI for such large systems requires a lot of data

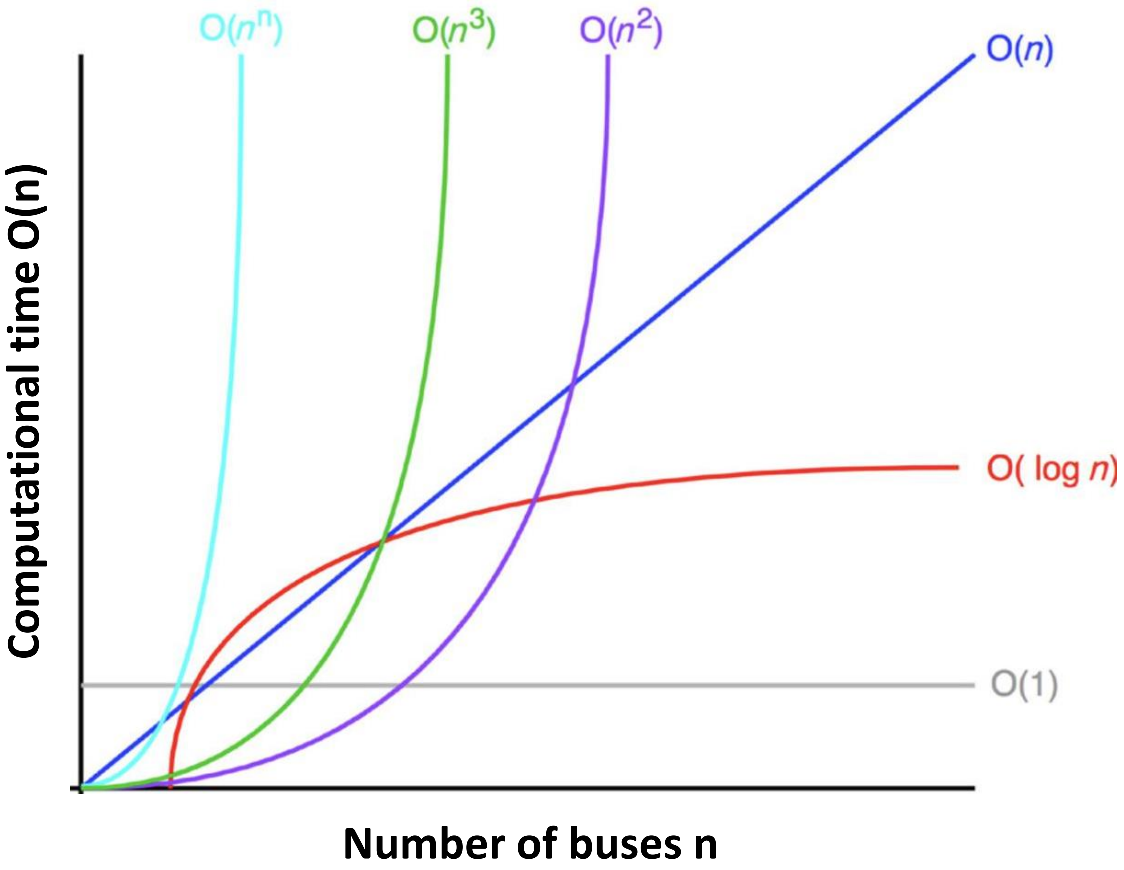


3. **AI does not provide explanations or robustness**
   - Energy systems are critical and sensitive infrastructures

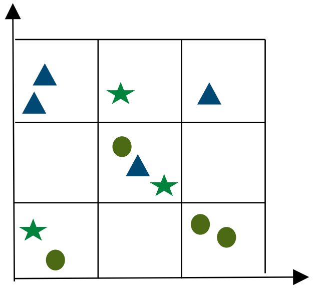
TU Delft

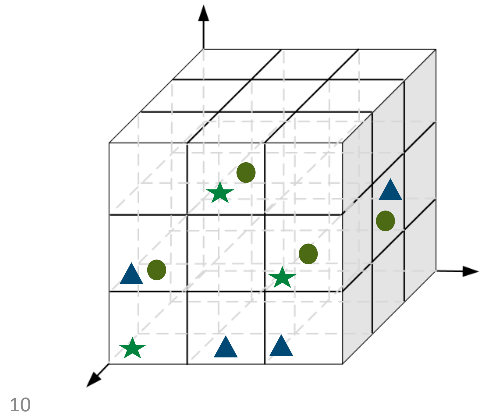
ML-driven ‘proxies’ for power system analysis

Challenge: scaling to system size



Curse of dimensionality





?

1d: 3 regions

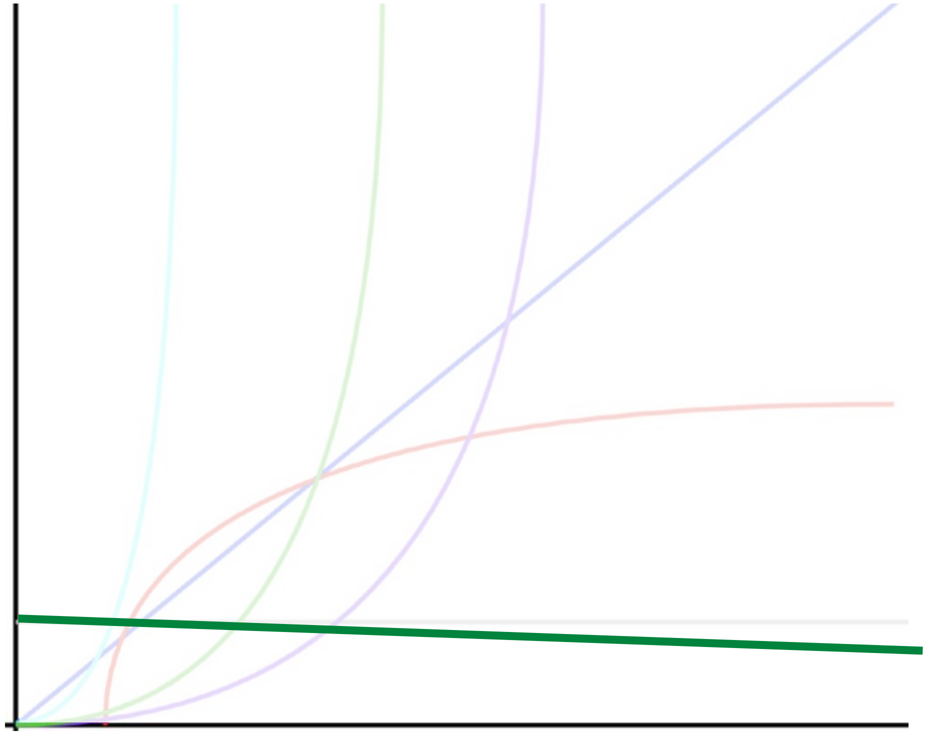
2d: 3^2 regions

3d: 3^3 regions

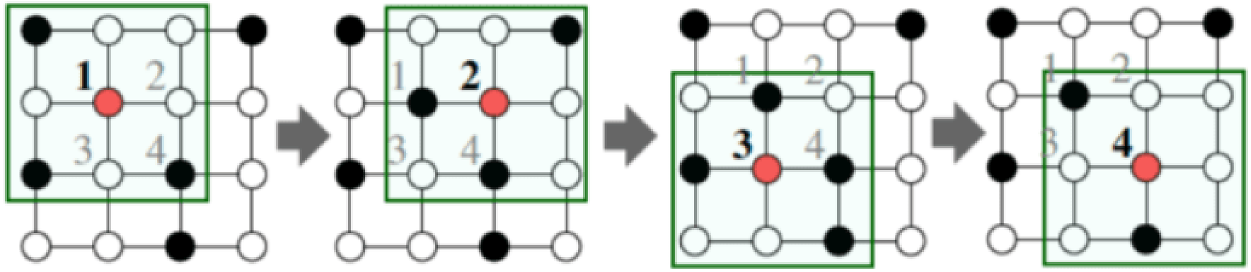
As dimensionality grows: fewer samples per region.

Learning without labels and inducing bias

- Self-supervised learning
- Physics-informed ML
- Weakly-supervised learning



Utopia?



Graph neural networks
learn from the neighbours

Distribution system state estimation

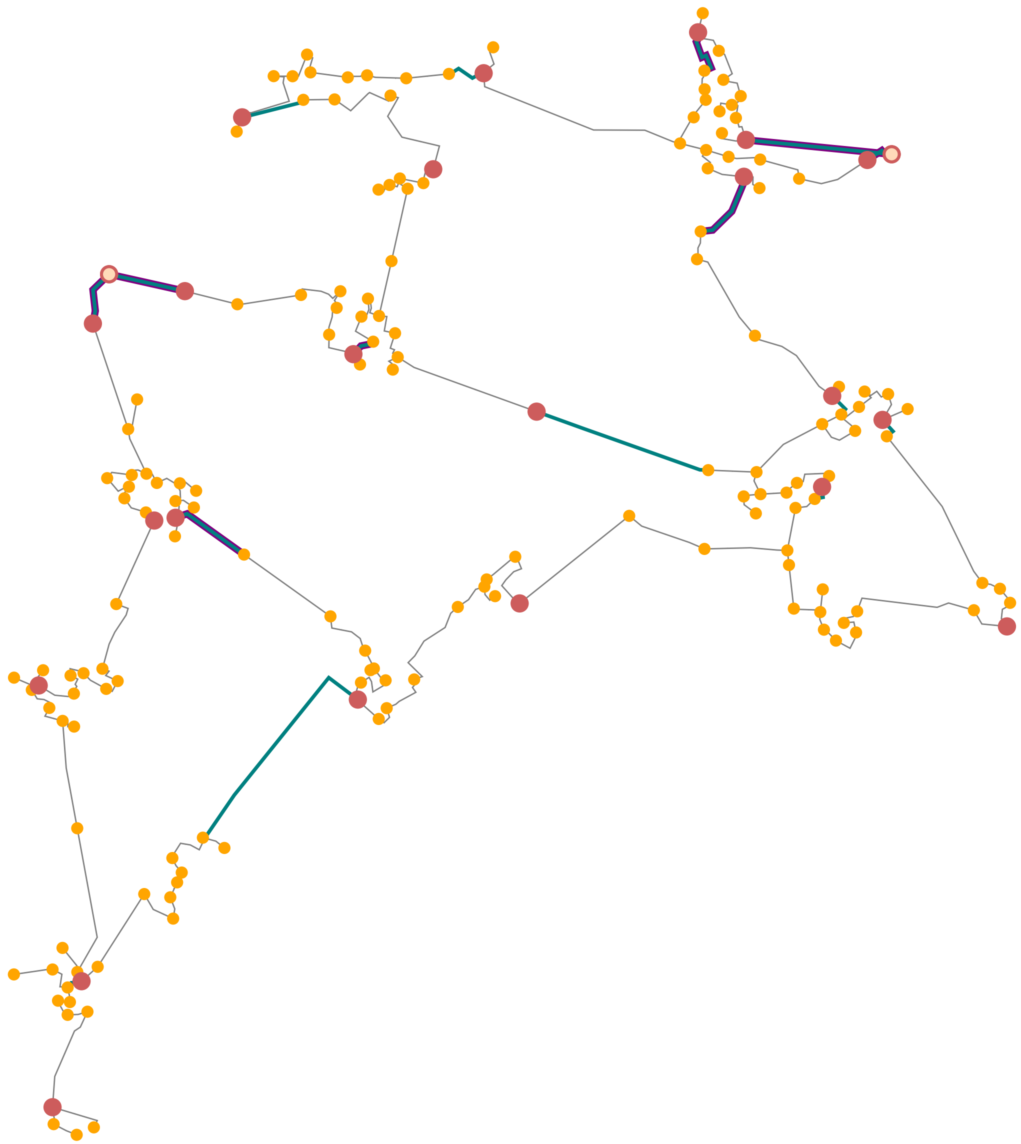
- Measurements $z \in \mathbb{R}^m$ with noise $\varepsilon \in \mathbb{R}^m$
- System state $x \in \mathbb{R}^{2n}$
- State estimation $f(z) \rightarrow x$
- Challenge: partial observable, scarce measurements $m \ll n$

TU Delft

A. Primadianto and C. -N. Lu, "A Review on Distribution System State Estimation," in *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3875-3883, Sept. 2017, doi: 10.1109/TPWRS.2016.2632156.

Distribution system (MV)

- Trafo
- Lines
- MV/LV buses
- HV buses
- Power flow measurement
- Voltage measurements

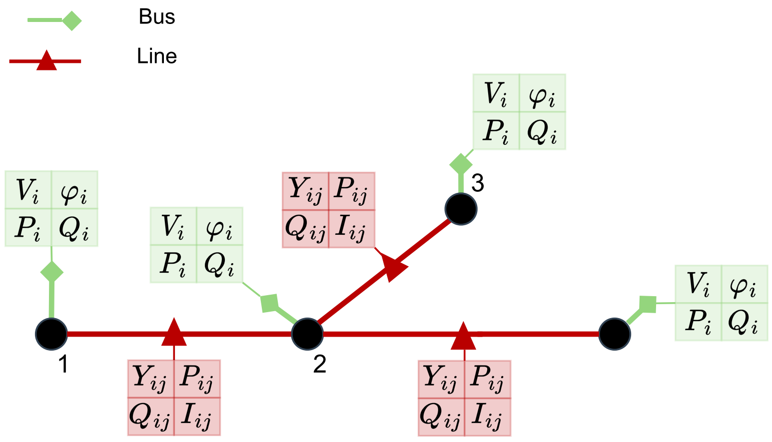


Model of the power flow

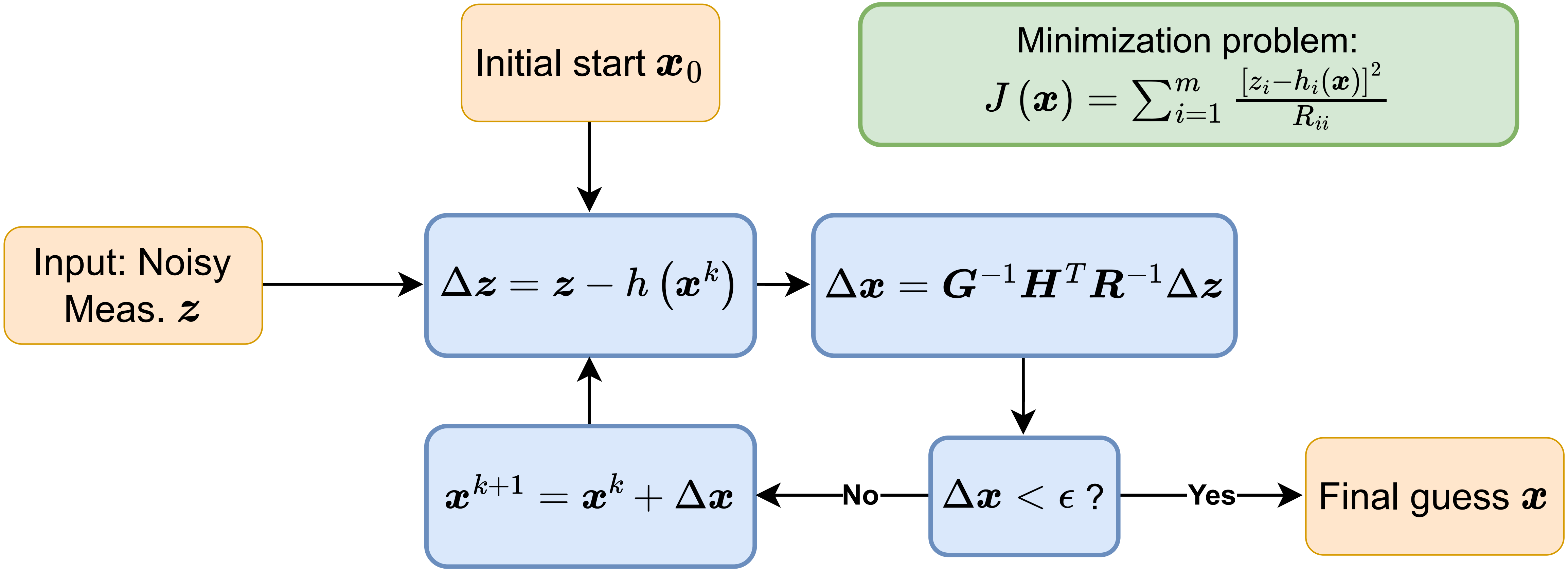
$$x = [V, \varphi]$$

$$z = h(x) + \varepsilon$$

$$h(x) = \begin{cases} V_i = V_i \\ \varphi_i = \varphi_i \\ P_{ij\rightarrow} = -V_iV_j[\mathbb{R}(Y_{ij})\, cos\, \Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\, sin\, \Delta\varphi_{ij}]] + V_i^2\left[\mathbb{R}(Y_{ij}) + \frac{\mathbb{R}(Y_{sij})}{2}\right] \\ P_{ij\leftarrow} = V_iV_j[-\mathbb{R}(Y_{ij})\, cos\, \Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\, sin\, \Delta\varphi_{ij}]] + V_j^2\left[\mathbb{R}(Y_{ij}) + \frac{\mathbb{R}(Y_{sij})}{2}\right] \\ Q_{ij\rightarrow} = V_iV_j[-\mathbb{R}(Y_{ij})\, \sin \Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\, \cos \Delta\varphi_{ij}]] - V_i^2\left[\mathbb{I}(Y_{ij}) + \frac{\mathbb{I}(Y_{sij})}{2}\right] \\ Q_{ij\leftarrow} = V_iV_j[\mathbb{R}(Y_{ij})\, sin\, \Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\, cos\, \Delta\varphi_{ij}]] - V_j^2\left[\mathbb{I}(Y_{ij}) + \frac{\mathbb{I}(Y_{sij})}{2}\right] \\ I_{ij\rightarrow} = -\left|\frac{P_{ij\rightarrow} - jQ_{ij\rightarrow}}{\sqrt{3}V_i e^{-j\varphi_i}}\right| \\ I_{ij\leftarrow} = -\left|\frac{P_{ij\leftarrow} - jQ_{ij\leftarrow}}{\sqrt{3}V_j e^{-j\varphi_j}}\right| \\ P_i = -\sum_{j\in N_x(i)} P_{ij\leftarrow} + P_{ij\rightarrow} \\ Q_i = -\sum_{j\in N_x(i)} Q_{ij\leftarrow} + Q_{ij\rightarrow} \end{cases}$$



Weighted least squares method



How accurate is x?
Convergence?

TU Delft

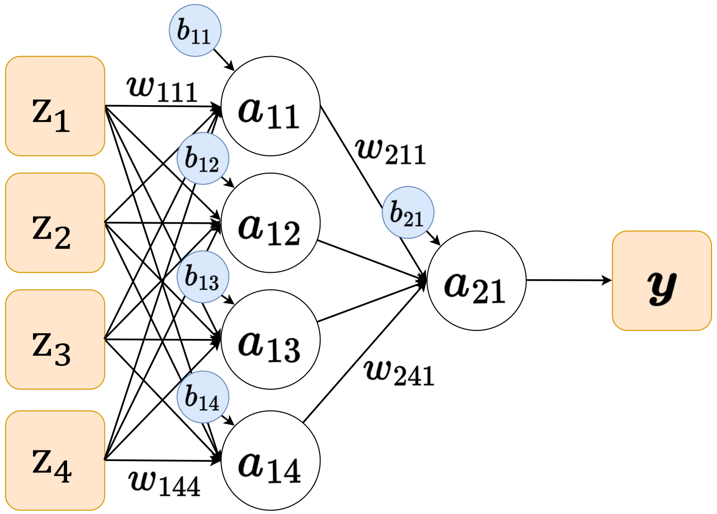
Statistical learning

- Training set $S = \{(z_1, y_1), (z_2, y_2) \dots (z_t, y_t)\}$ with t samples

- Inference problem is to find a function $f: Z \rightarrow Y$ such that $f(z) \sim y$

- Common loss function $\mathrm{L}(f(z), y)$ for regression is the square loss

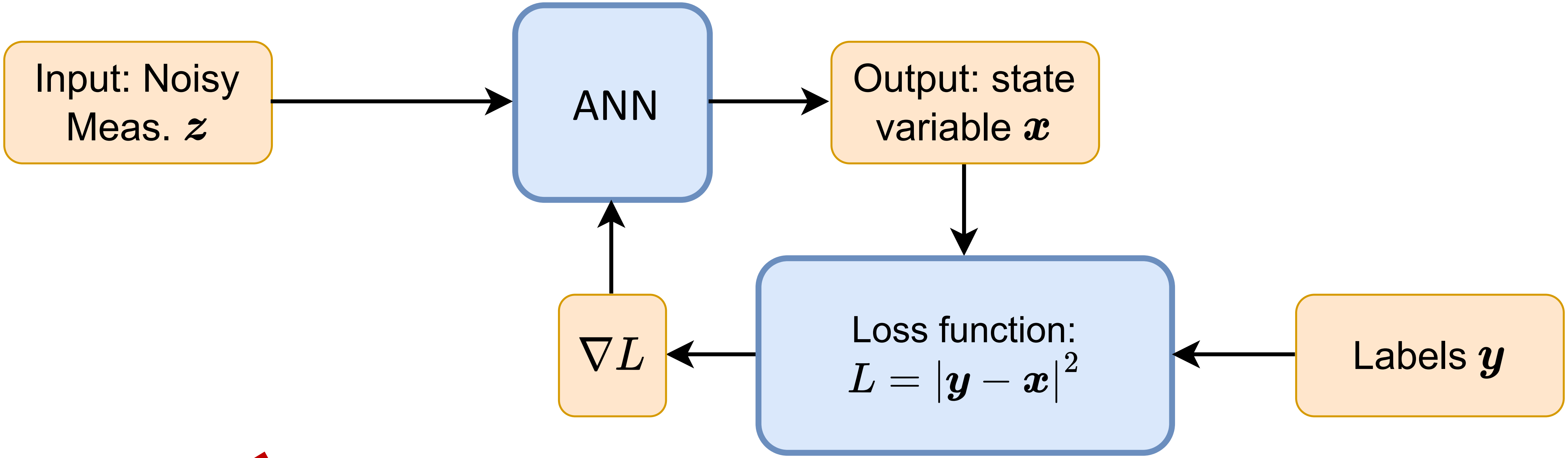
Artificial Neural Network (ANN)



$$f_\theta: Z \rightarrow Y$$

Note: this notation differs to conventional statistical learning where x is feature

Supervised learning for state estimation



Labels are not known

Newton's method generates label with "errors" $\hat{y} = y + \gamma^N$

Weakly-supervised learning

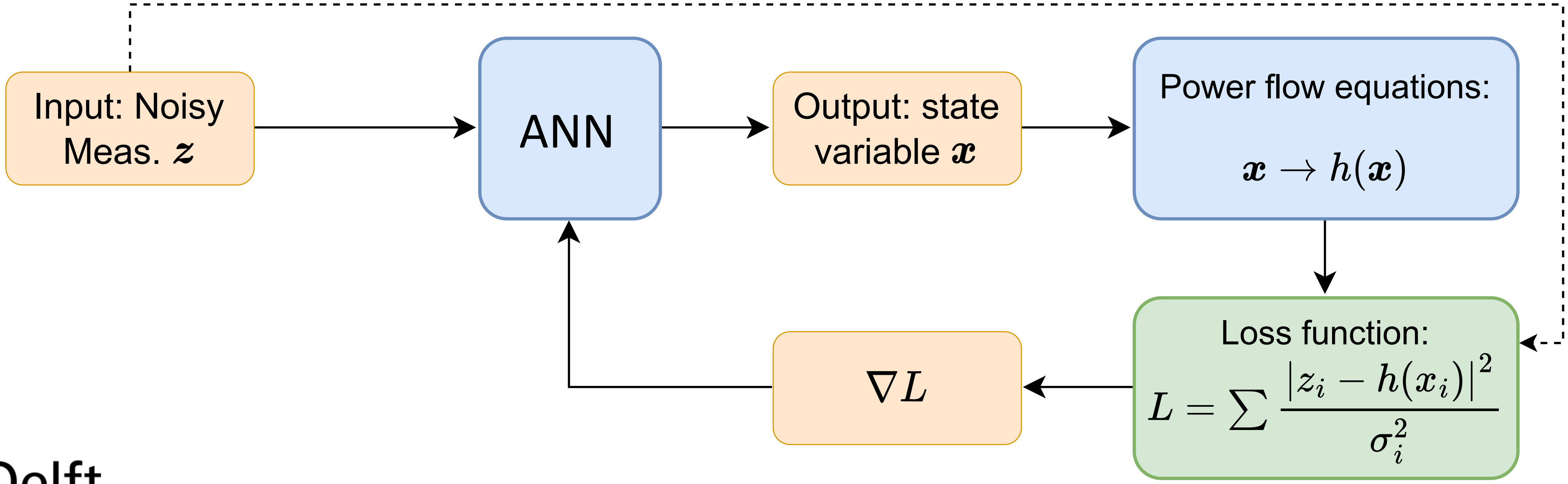
- Inaccurate input and output
- Learn with inaccurate labels $S = \{(z_1, \hat{y}_1), (z_2, \hat{y}_2) \ldots (z_t, \hat{y}_t)\}$
- Design a loss function $\mathrm{L}(f(z), \hat{y})$
- Objective: learning $f: Z \rightarrow Y$ such that $f(z) \sim y$

TU Delft

Y. -F. Li, L. -Z. Guo and Z. -H. Zhou, "Towards Safe Weakly Supervised Learning," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 334-346, 1 Jan. 2021, doi: 10.1109/TPAMI.2019.2922396.

Weakly-supervised learning for state estimation

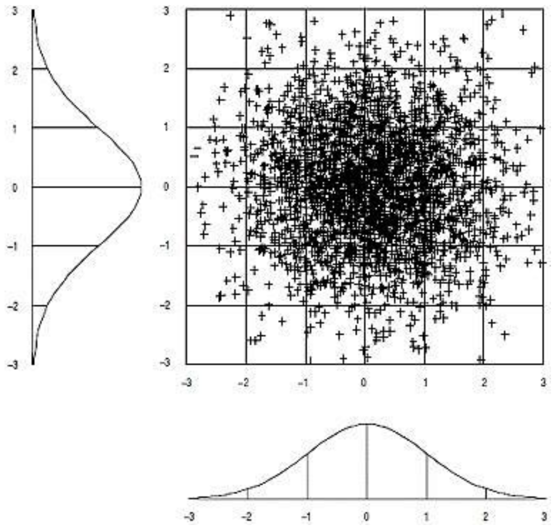
- ANN $f(z) \rightarrow x$
- Measurement function using power flow equations $\mathrm{h}(\mathrm{x}) \rightarrow \hat{z}$



TU Delft

Respect the structure of the domain

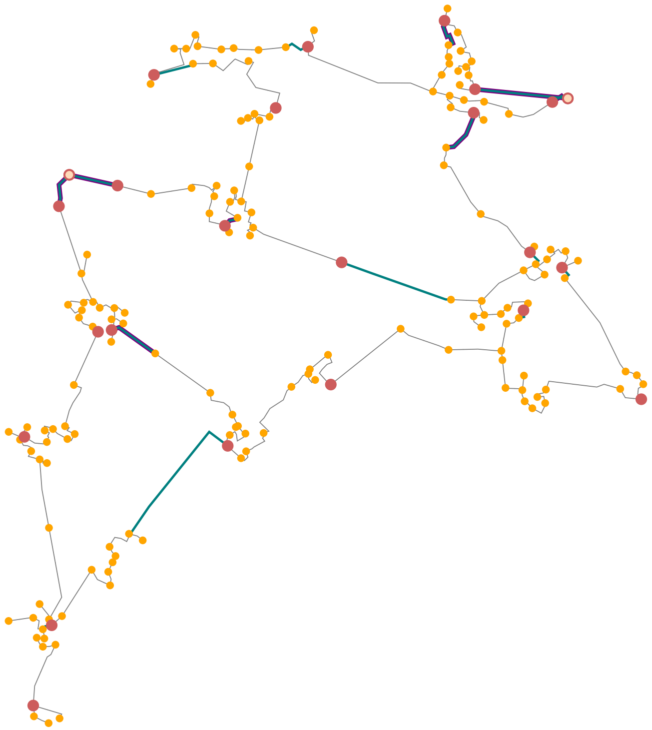
Noisy measurements



Power flow equations

$$h(x) = \begin{cases} V_i = V_i \\ \varphi_i = \varphi_i \\ P_{ij\rightarrow} = -V_i V_j\left[\mathbb{R}(Y_{ij})\cos\Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\sin\Delta\varphi_{ij}\right] + V_i^2\left[\mathbb{R}(Y_{ij}) + \frac{\mathbb{R}(Y_{sij})}{2}\right] \\ P_{ij\leftarrow} = V_i V_j\left[-\mathbb{R}(Y_{ij})\cos\Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\sin\Delta\varphi_{ij}\right] + V_j^2\left[\mathbb{R}(Y_{ij}) + \frac{\mathbb{R}(Y_{sij})}{2}\right] \\ Q_{ij\rightarrow} = V_i V_j\left[-\mathbb{R}(Y_{ij})\sin\Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\cos\Delta\varphi_{ij}\right] - V_i^2\left[\mathbb{I}(Y_{ij}) + \frac{\mathbb{I}(Y_{sij})}{2}\right] \\ Q_{ij\leftarrow} = V_i V_j\left[\mathbb{R}(Y_{ij})\sin\Delta\varphi_{ij} + \mathbb{I}(Y_{ij})\cos\Delta\varphi_{ij}\right] - V_j^2\left[\mathbb{I}(Y_{ij}) + \frac{\mathbb{I}(Y_{sij})}{2}\right] \\ I_{ij\rightarrow} = -\left|\frac{P_{ij\rightarrow} - jQ_{ij\rightarrow}}{\sqrt{3}V_i e^{-j\varphi_i}}\right| \\ I_{ij\leftarrow} = -\left|\frac{P_{ij\leftarrow} - jQ_{ij\leftarrow}}{\sqrt{3}V_j e^{-j\varphi_j}}\right| \\ P_i = -\sum_{j\in N_x(i)} P_{ij\leftarrow} + P_{ij\rightarrow} \\ Q_i = -\sum_{j\in N_x(i)} Q_{ij\leftarrow} + Q_{ij\rightarrow} \end{cases}$$

Topology



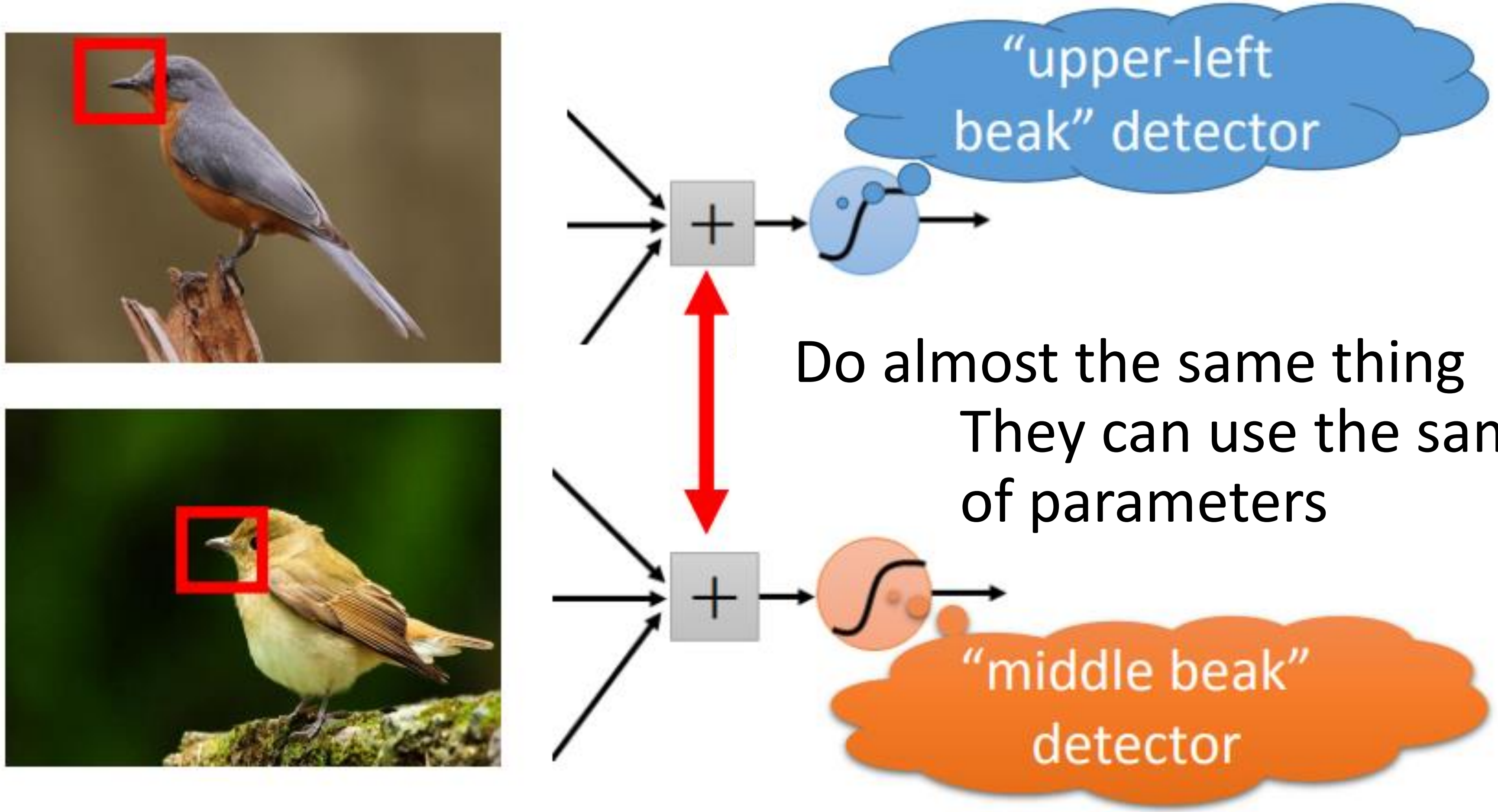
TU Delft

AI successes in the domain ‘images’

Property 1: Some patterns are much smaller than the whole image. A neuron does not have to see the whole image to discover the pattern.

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

Property 2: The same patterns appear in different regions. (translated invariance)



Do almost the same thing

They can use the same set of parameters

TU Delft

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

CNN— Convolution layer

Stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
0	0	0	0	1	0
1	1	1	0	1	0
0	0	0	0	1	0

6×6 image

Those are the network parameters to be learned

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1 Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2 Matrix

3	-1	-3	-1
-2	1	-1	-3
-2	-4	0	1
-1	0	-2	-1

Property 1

Each filter detects a small pattern (3 x 3)

James, Gareth, et al. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.

CNN— Convolution layer

Stride=1

0	1	0	0	0	1
0	1	0	0	1	0
0	0	0	1	0	0
0	0	0	1	0	0
1	1	1	0	1	0
0	0	0	0	0	1

6×6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1
Matrix

0	1	-2	-1
-1	1	-1	-3
-1	-4	0	-1
-1	-1	-3	3

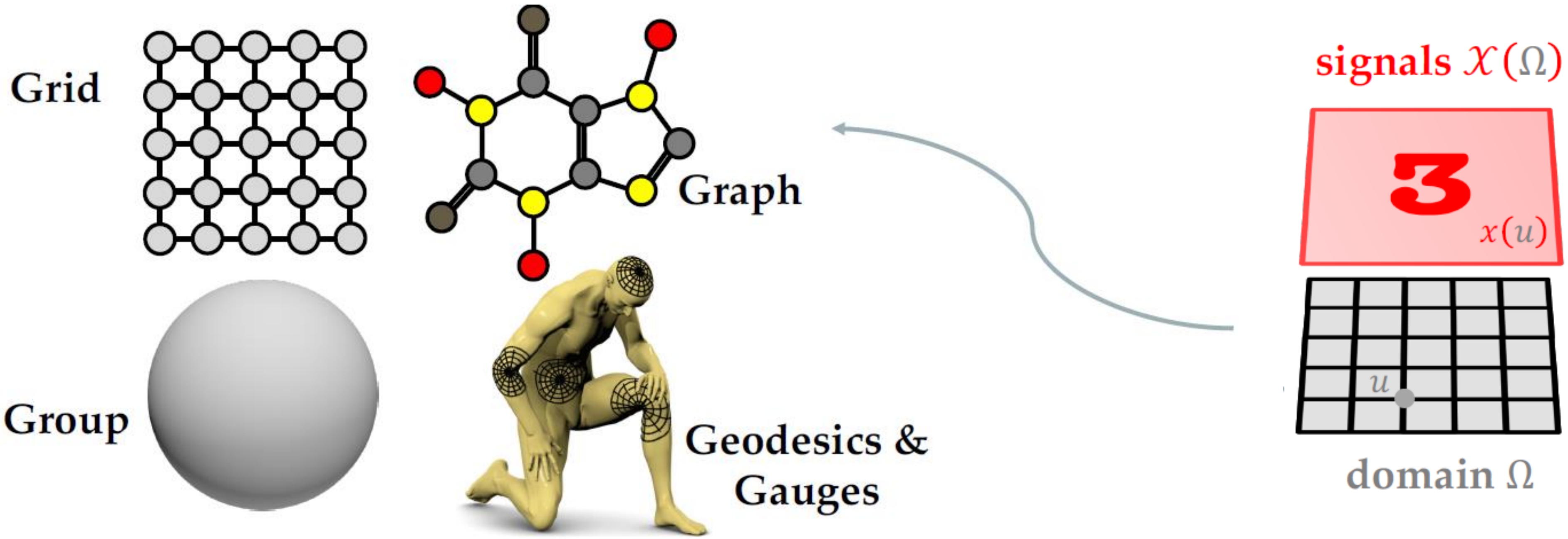
Property 2

The same patterns that appear in different regions can be detected.

Not much image-like data in power system operation and planning…

Geometric deep learning

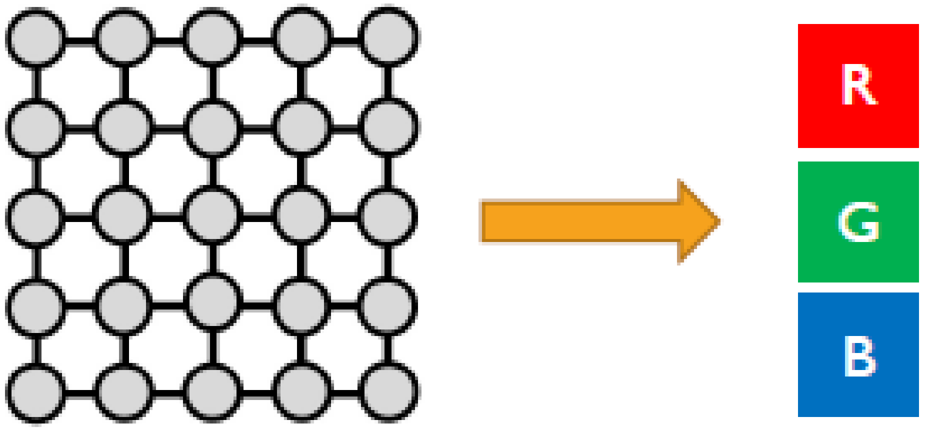
- Data are signals x on geometric domains Ω
- A signal x on Ω is a function $\chi(\Omega, C) = \{x: \Omega \to C\}$



Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Graph Neural Networks (GNNs)

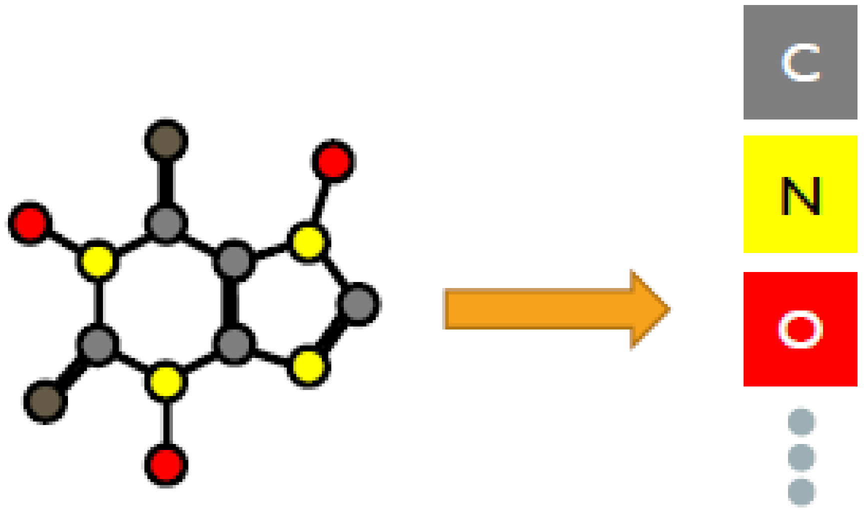
Example: $p \times p$ RGB image



$\Omega = \mathbb{Z}_p \times \mathbb{Z}_p$

$\mathcal{C} = \mathbb{R}^3$

Example: molecular graph



$\Omega = \{1, \ldots n\}$

$\mathcal{C} = \mathbb{R}^c$

TU Delft

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

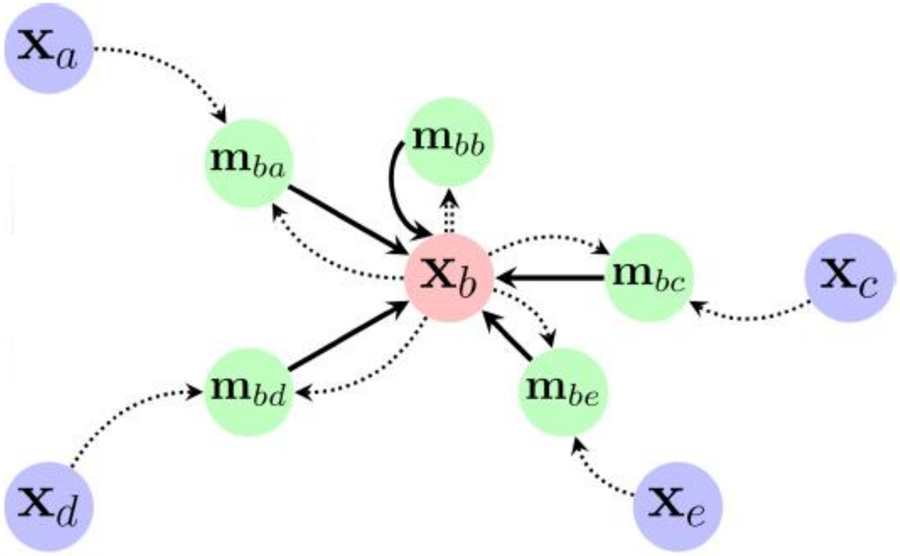
Locality on graphs: Neighbourhoods

- Consider graph $G = (V, E)$ where $E \subseteq V \times V$
- Adjacency matrix A with

$$a_{ij} = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E \end{cases}$$

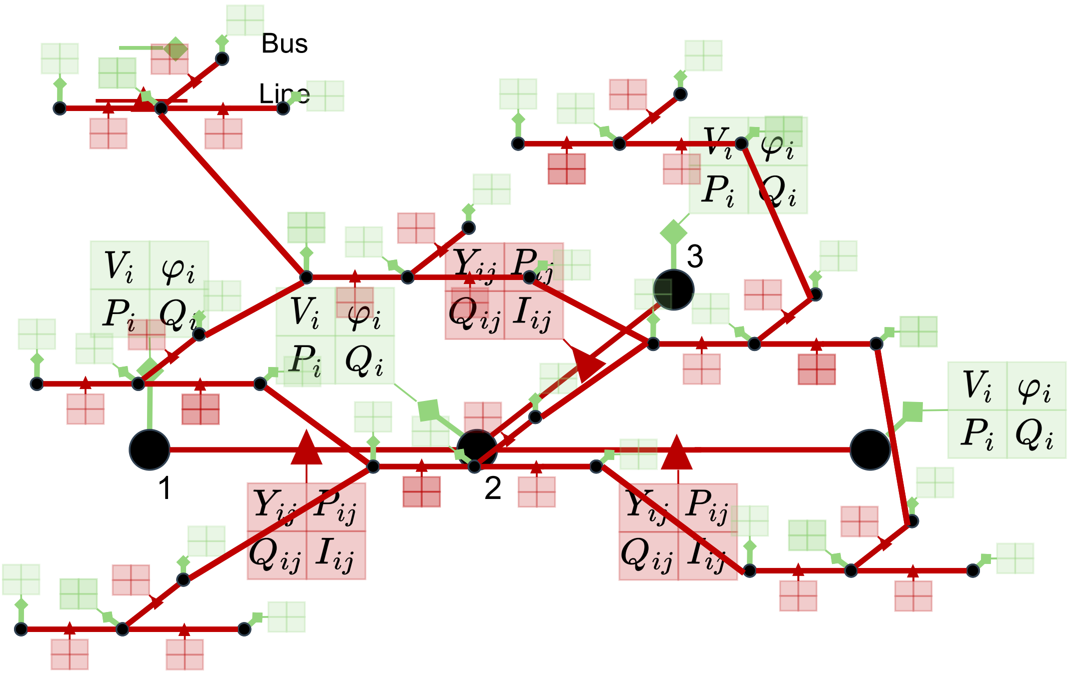
- (1-hop) neighbourhood $N_i = \{j: (i,j) \in E \cup (j,i) \in E\}$ for a node i
- Neighbourhood features $\boldsymbol{X}_{N_i} = \{\{x_j: j \in N_i\}\}$
- Local function, $\phi(x_i, \boldsymbol{X}_{N_i})$, operating over them.

Convolutional layers & message passing



$$\eta_i = \phi\left(x_i, \bigoplus_{j \in N_i} \psi(x_i, x_j)\right)$$

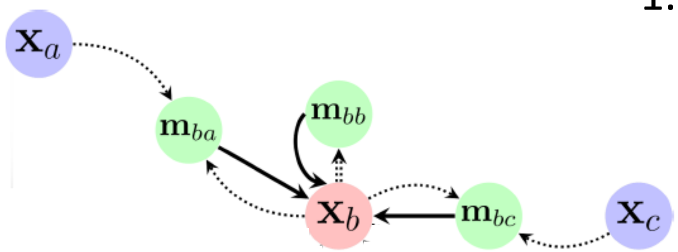
State estimation



Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... & Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv e-prints*, arXiv-1806.

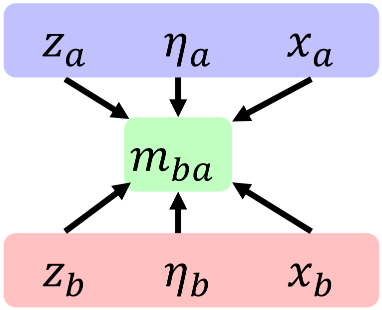
TU Delft

Deep statistical solver



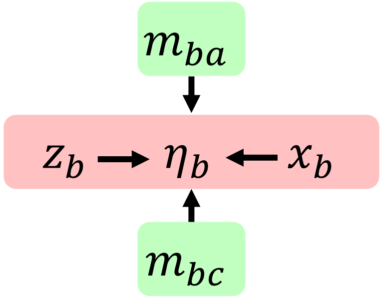
0. Initialize $x = x^0, \eta = \eta^0$

1. update edges



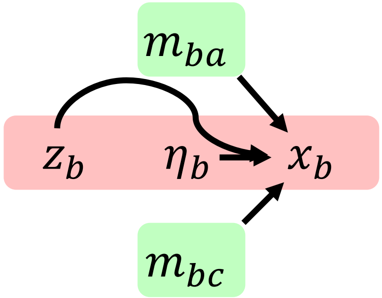
$$m_{ba} \leftarrow m_{ba} + \Delta t \times \left[\phi_\theta^{ba}(t, z_a, \eta_a, x_a) + \phi_\theta^{ba}(t, z_b, \eta_b, x_b)\right]$$

2. update vertices



$$\eta_b \leftarrow \eta_b + \Delta t \times \phi_\theta^{b}(t, z_b, \eta_b, x_b, m_{ba}, m_{bc}, m_{bb})$$

3. update label



$$x_b \leftarrow x_b + \Delta t \times \phi_\theta^{bx}(t, z_b, \eta_b, x_b, m_{ba}, m_{bc}, m_{bb})$$

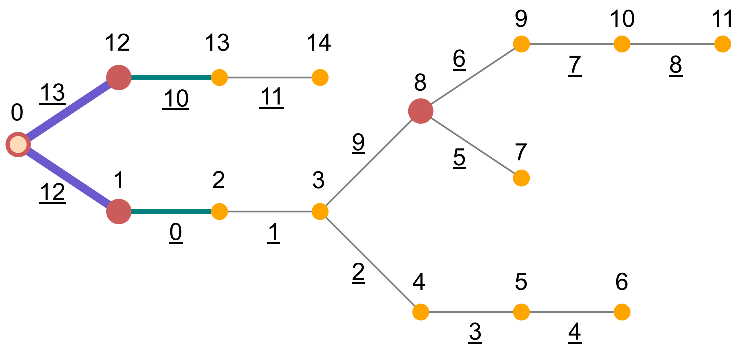
Perform $t = 1, \dots T$

TU Delft

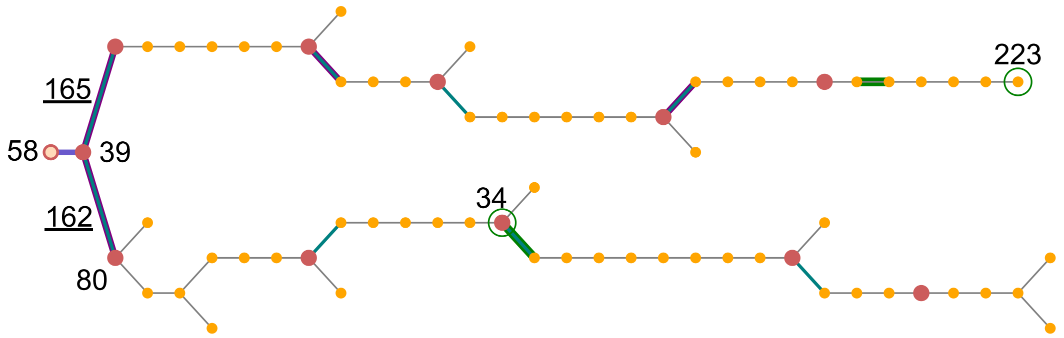
Balthazar Donon, Liu, Z., Liu, W., Guyon, I., Marot, A., & Schoenauer, M. (2020). Deep statistical solvers. *Advances in Neural Information Processing Systems*, *33*, 7910-7921.

Case study: power systems

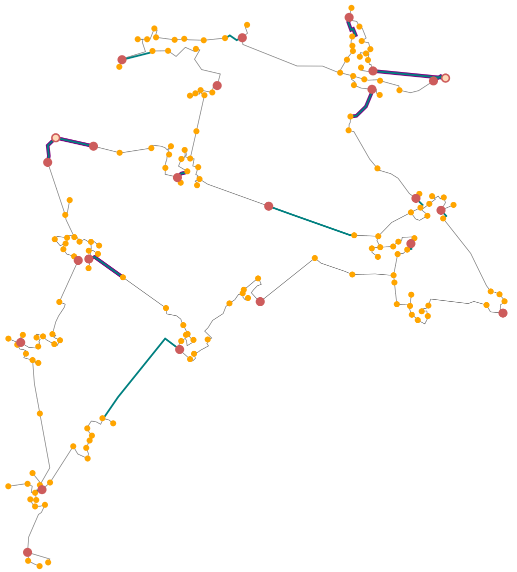
14-bus CIGRE MV grid from



70-bus Oberrhein MV sub-grid from



179-bus Oberrhein MV grid from



- Trafo
- Lines
- MV/LV buses
- HV buses
- Power flow measurement
- Voltage measurements
- Focus bus

B. Habib, E. Isufi, W. v. Breda, A. Jongepier and J. L. Cremer, "Deep Statistical Solver for Distribution System State Estimation," in IEEE Transactions on Power Systems, doi: 10.1109/TPWRS.2023.3290358.

Case study settings

Data generation

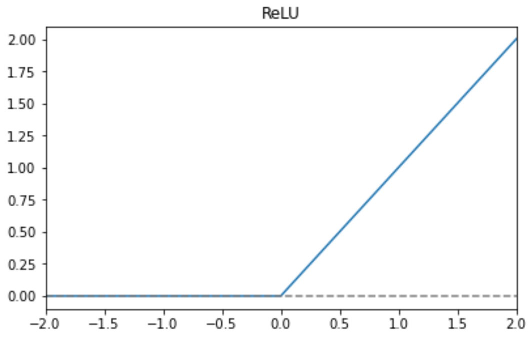
- 8640 days, with each 24 hours, +/- 15% around Gaussian on loads
- Balanced system, pandapower, AC power flow
- Measurement noise
  - $0.5\% - 2\%$ for the voltage and current measurement
  - $1\% - 5\%$ for the active and reactive power measurement
  - Pseudomeasurement were generic load profiles
- Baselines
  - Weighted least square (WLS)
  - Feedforward Neural Network (FFNN)
  - supervised DSS^2

Model & hyperparameters

- Hyper-Heterogeneous Multi GNN
- Training 80%, validation 10%, testing 10%
- Grid search on learning rate λ, layer dimensions d, and layer numbers

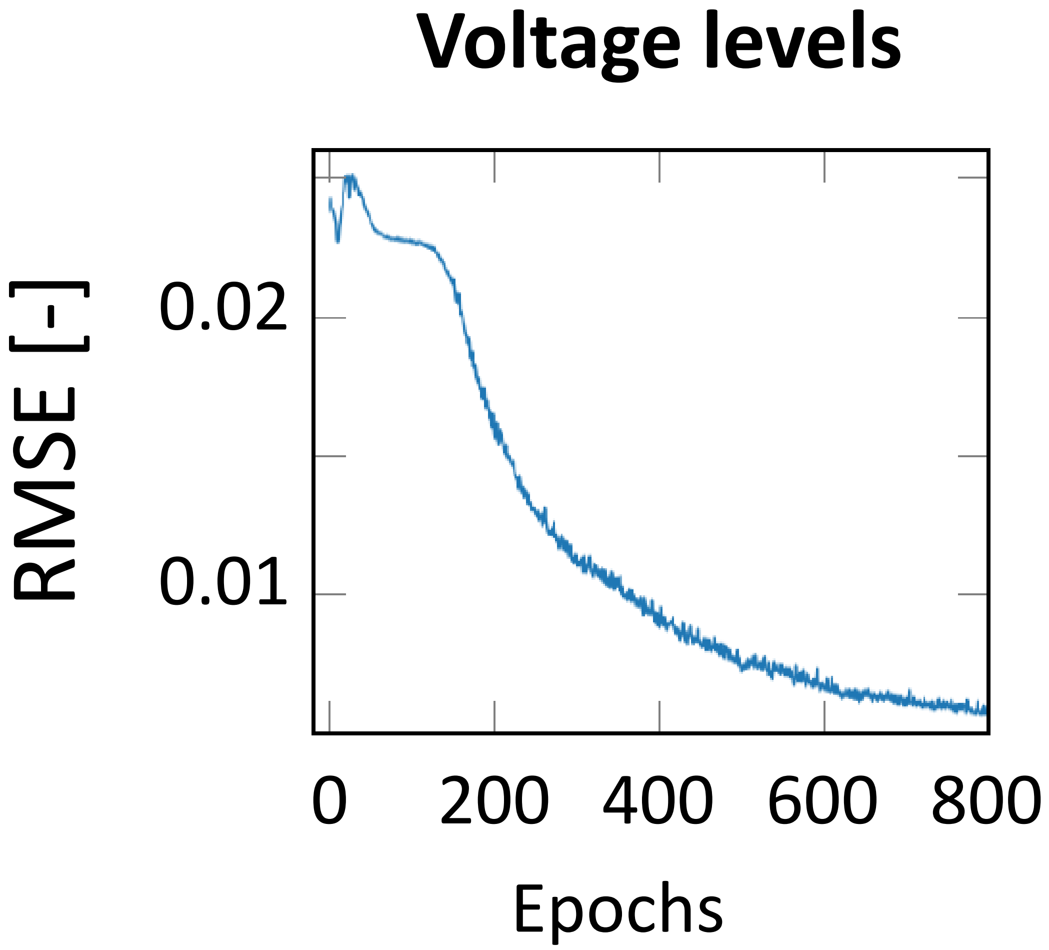
Assume stable system

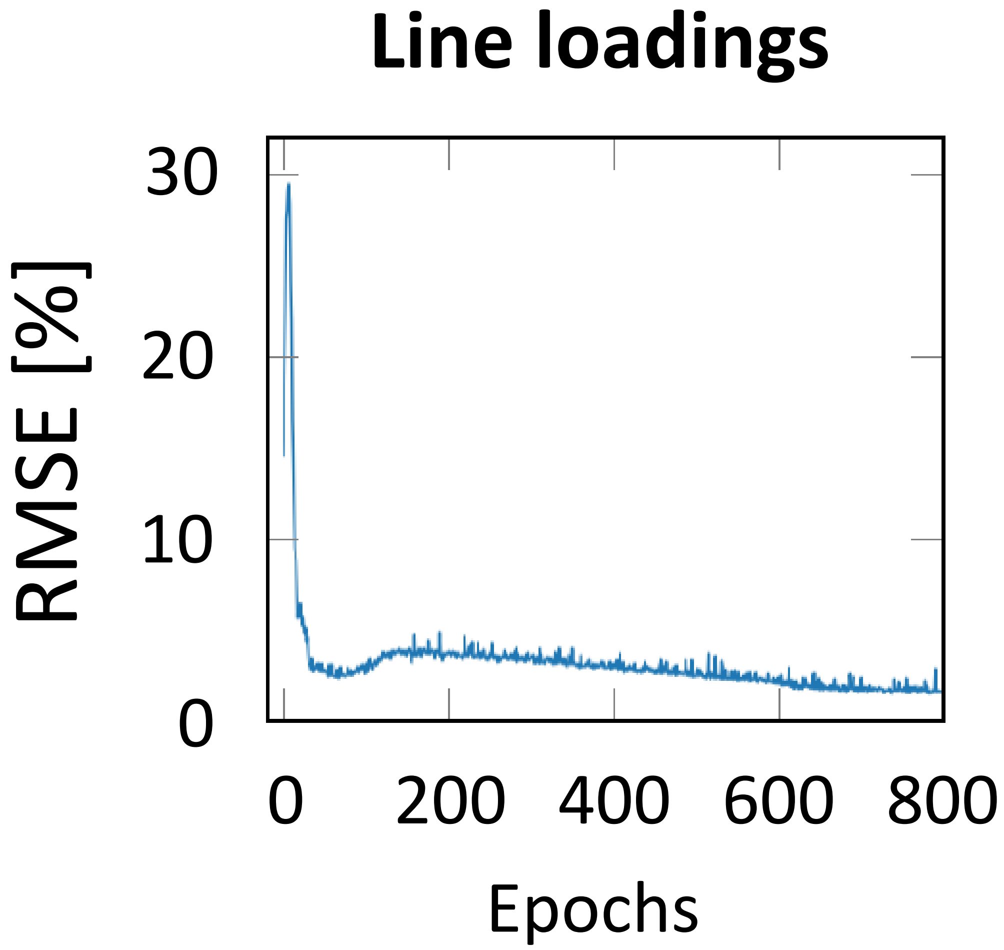
$$L(\boldsymbol{z}, \boldsymbol{x}) = \sum_{i \in m} \frac{|z_i - h_i(\boldsymbol{x})|^2}{R_{ii}} + \lambda[\text{ReLU}(V - 1.05) + \text{ReLU}(0.95 - V) + \text{ReLU}(\text{ loading } - 100) + \text{ReLU}(\varphi - 0.25) + \text{ReLU}(-0.25 - \varphi)]$$



TU Delft

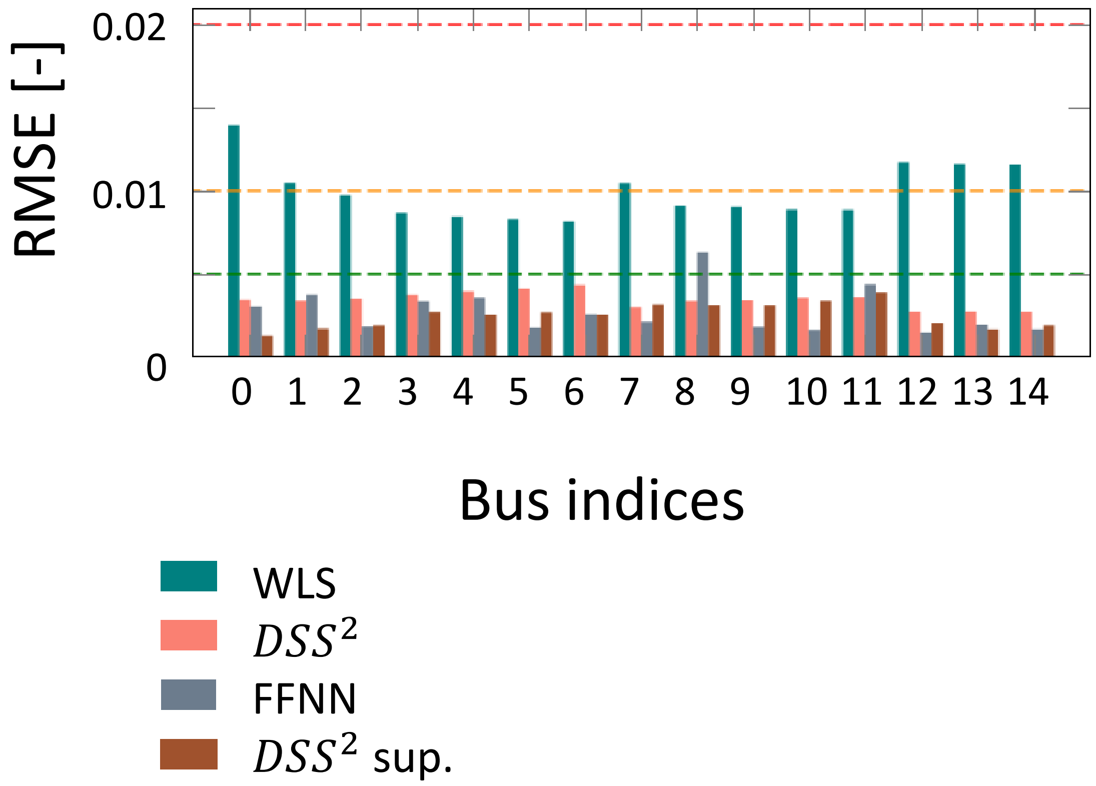
Training performance 14-bus system



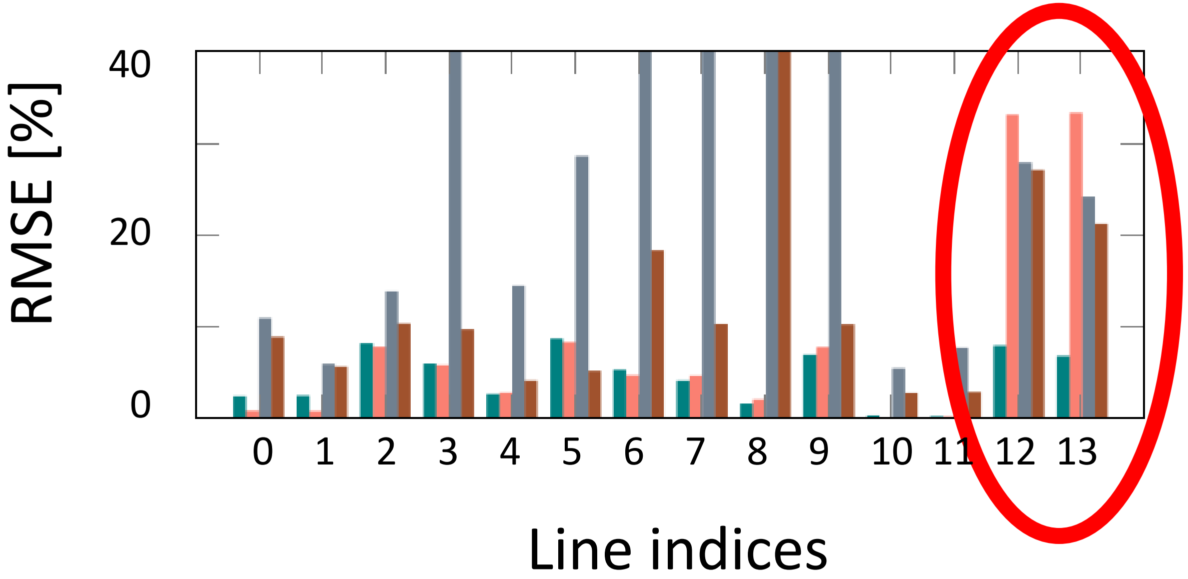


State estimation 14-bus system

Voltage levels



Line loadings



Model inaccuracies: assumed transformer = lines!

Accuracy

	14-bus system			
Metric \ approach	WLS	ANN	sup DSS^2	DSS^2
Voltage RMSE $[10^{-3}]$	10	3	**3**	3
Line loading RMSE [%]	**3**	42	13	4
Trafos loading RMSE [%]	**5**	39	14	8

* with increased convergence rate
** with higher tolerance and more iterations

Convergence

	14-bus system				70-bus Oberrhein			179-bus Oberrhein	
Metric \ approach	WLS	ANN	sup DSS^2	DSS^2	WLS	WLS*	DSS^2	WLS**	DSS^2
Voltage RMSE $[10^{-3}]$	10	3	3	3	31	6	2	10	2
Line loading RMSE [%]	3	42	13	4	17	15	2	6	3
Trafos loading RMSE [%]	5	39	14	8	39	24	3	4	4
Convergence [%]	100	100	100	100	25	100	100	53	100

- WLS did not converge in some instances (25%-50%)
- DSS^2 always ‘converges’ (produces a label)

* with increased convergence rate
** with higher tolerance and more iterations

Computational 'prediction' time [ms]

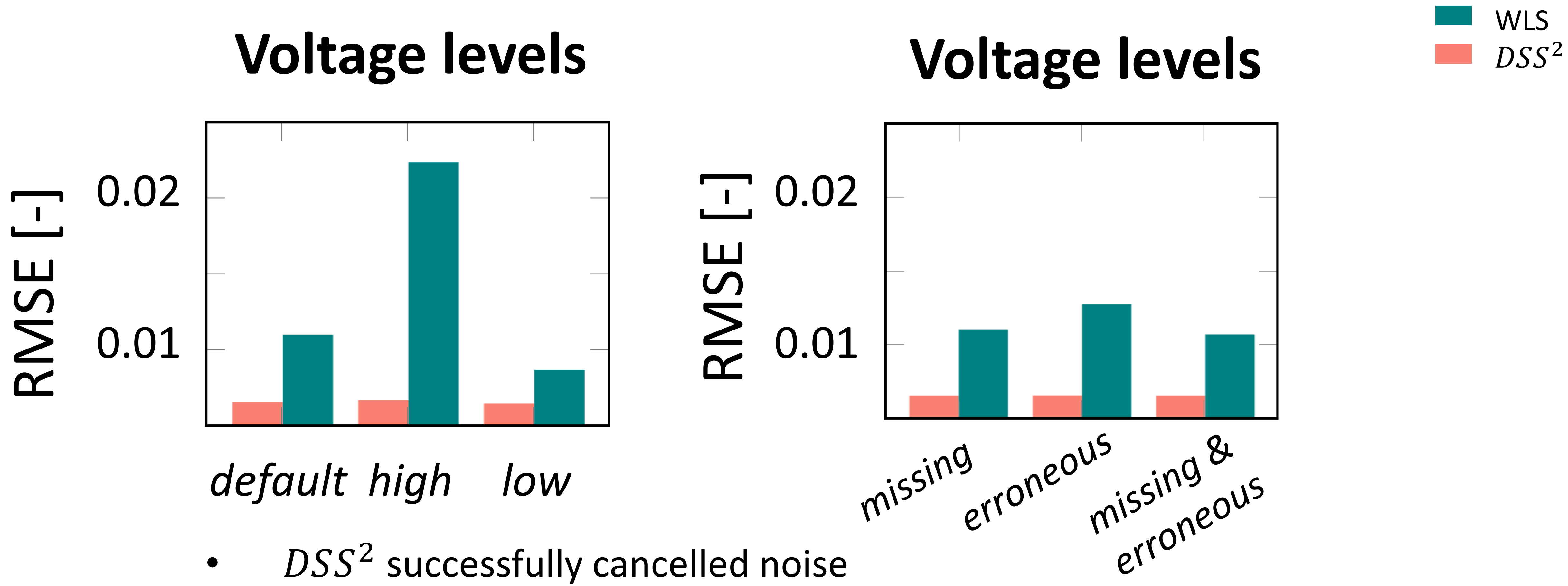
	14-bus system				70-bus Oberrhein			179-bus Oberrhein	
Metric \ approach	WLS	ANN	sup DSS^2	DSS^2	WLS	WLS*	DSS^2	WLS**	DSS^2
Voltage RMSE $[10^{-3}]$	10	3	3	3	31	6	2	10	2
Line loading RMSE [%]	3	42	13	4	17	15	2	6	3
Trafos loading RMSE [%]	5	39	14	8	39	24	3	4	4
Convergence [%]	100	100	100	100	25	100	100	53	100
Computational time $[ms]$	86	**4**	5	6	123	161	**26**	1212	**58**

~10 **~2**

- WLS increases significantly with system size
- DSS^2 increases moderately with system size

* with increased convergence rate
** with higher tolerance and more iterations

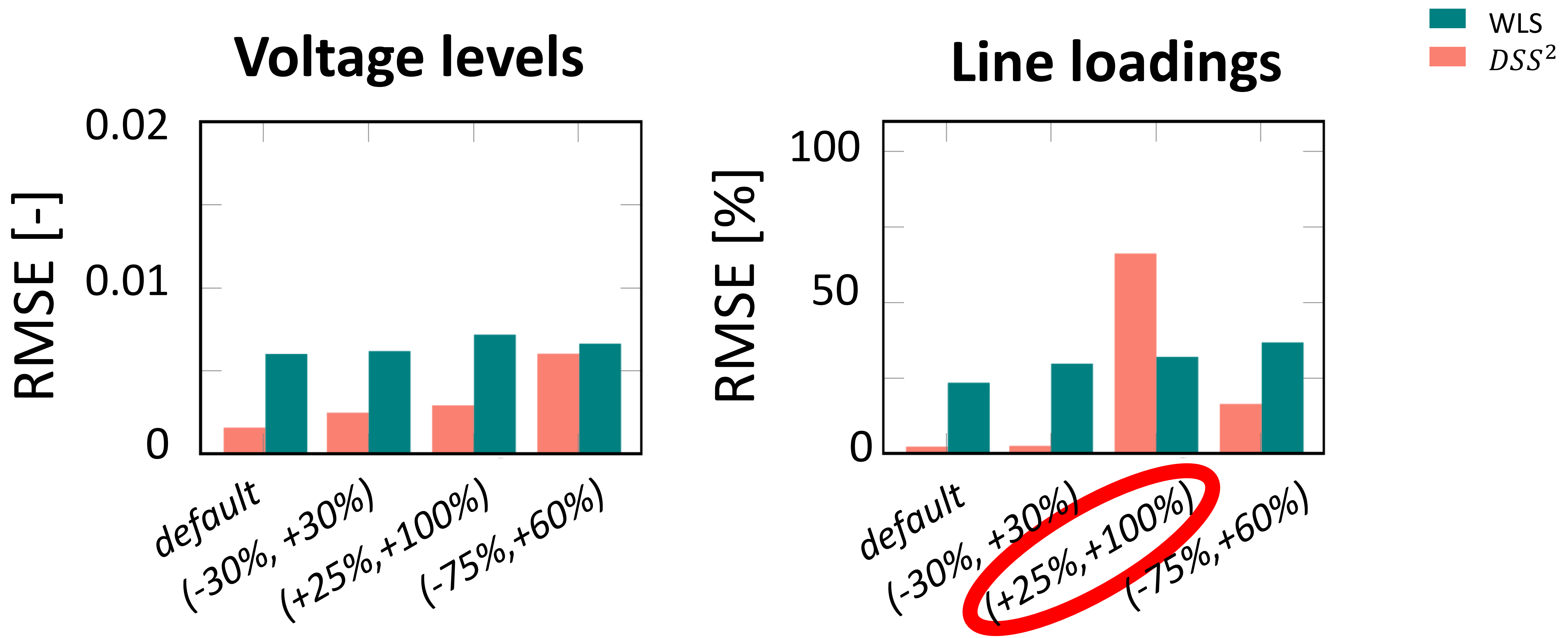
Noise and missing, erroneous data



- DSS^2 successfully cancelled noise
- DSS^2 was not trained to handle such events
- GNN architecture increased the interpolation capabilities by incorporating the data symmetries w.r.t. the underlying graph

TUDelft

Increase in (generation, load)



Limitation: high load levels

TUDelft

Drivers

- Combining interdisiplinary approaches from statistics, machine learning and mathematical optimization to a power system problem
- Learning without labels and with structural information (inducing bias, regularising with physics, etc)

Generic barriers

- Curse of dimensionality
- Performance on out-of-distribution
- Guarantees

Challenges applying GNNs to power systems

- Power systems are not always well-meshed graphs
- Enforcing power flow equations
- Addressing nonlinearity -> multiple solutions
- Graph topology changes (dynamic graphs)
- …

Contact and references

Speaker

Jochen Cremer

Assistant Professor IEPG, TU Delft

Web: https://www.tudelft.nl/ai/delft-ai-energy-lab

Email: j.l.cremer@tudelft.nl

Collaborators


Benjamin Habib


Elvin Isufi


Ward van Breda


Arjen Jongepier

STEDIN.NET

Reference

- B. Habib, E. Isufi, W. van Breda, A. Jongepier and J. L. Cremer, "Deep Statistical Solver for Distribution System State Estimation," in *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 4039-4050, March 2024, doi: 10.1109/TPWRS.2023.3290358. **[open access]**
- Code: https://github.com/TU-Delft-AI-Energy-Lab

TU Delft

Related references & code

Workshop on AI for Energy Systems: https://github.com/TU-Delft-AI-Energy-Lab/Workshop_AI_for_Intelligent_Energy_Systems

Weakly supervised learning for power systems (**Example code:** https://github.com/TU-Delft-AI-Energy-Lab/Deep-Statistical-Solver-for-Distribution-System-State-Estimation)

- Bastien Giraud, Ali Rajaei, Jochen L. Cremer "Constraint-Driven Deep Learning for N-k Security Constrained Optimal Power Flow", *Electric Power System Research and 2024 IEEE Power System Computation Conference*
- B. Habib, E. Isufi, W. v. Breda, A. Jongepier and Jochen L. Cremer, "Deep Statistical Solver for Distribution System State Estimation, " *IEEE Transactions on Power Systems, 2023, doi: 10.1109/TPWRS.2023.3290358*.

Cost-sensitive learning

- Dariush Wahdany, Carlo Schmitt, Jochen L. Cremer, "More than Accuracy: End-To-End Wind Power Forecasting that Optimises the Energy System", *Electric Power System Research, 2023*
- A. Bugaje, J. L. Cremer, M. Sun, G. Strbac, "Selecting DT Models for Security Assessment using ROC- and Cost-Curves", *Energy and AI*, 2021: 100110.
- J. L. Cremer, G. Strbac, "A Machine-learning based Probabilistic Perspective on Dynamic Security Assessment" *International Journal of Electrical Power & Energy System, 2020*

Interpretable models (**Example code:** https://github.com/JochenC/From-optimization-based-machine-learning-to-interpretable-security-rules-for-operation)

- J. L. Cremer, I. Konstantelos, G. Strbac, "From Optimization-based Machine Learning to Interpretable Security Rules for Operation", *IEEE Transactions on Power Systems, 2019*
- J. L. Cremer, I. Konstantelos, S. H. Tindemans, G. Strbac, "Data-driven Power System Operation: Exploring the Balance between Cost and Risk", *IEEE Transactions on Power Systems, 2018*

Sampling with optimisation-based constraints (**Example code:** https://github.com/TU-Delft-AI-Energy-Lab/Split-based-sampling)

- Al-Amin Bugaje, Jochen L. Cremer, Goran Strbac "Generating Quality Datasets for Real-Time Security Assessment: Balancing Historically Relevant and Rare Feasible Operating Condition" *International Journal of Electrical Power & Energy Systems, 2023*

TU Delft

Thank you