Measurement-Driven and Physics-Informed Monitoring and Control of Power Systems

Misha Chertkov Applied Math @ UArizona

December 12, 2021 CDC 2021 Workshop: Uncertainty Management in Power System Dynamics

Applied Math '21 = Harvesting Data and Model Revolution

- Applied Math '21 = Traditional AM + Contemporary AM
- Traditional Applied Math
  - Natural Science Based (motivated by Physics, later Biology, Enviromental Sciences, etc ...)
  - Originally largely ODE, PDE, Dynamical Systems, Chaos, Turbulence, ...
- Contemporary AM
  - More applications, e.g. Engineering, Social sciences, Networks
  - AI disciplines: Statistics, Data Science, Computer Science, Machine Learning, Optimization, Control
  - Deep Learning - most prominent recent addition (automatic differentiation, very efficient large scale optimization) ... based a lot on "old" stuff (stochastic gradient descent, sensitivity analysis)

Trustworthy Scientific AI

Physics Informed Machine Learning

- Physics = exemplary of Quantitative Sciences (based on equations)
- Machine/Deep Learning = exemplary (arguably the hottest) AI discipline

Trustworthy Scientific AI

Scientific AI

- AI disciplines (foundational, empirical)
- + Hierarchy of AI Models (physics-agnostic) with "tunable" amount of "physics" added

Trustworthy Scientific AI

Scientific AI

- AI disciplines (foundational, empirical)
- + Hierarchy of AI Models (physics-agnostic) with "tunable" amount of "physics" added

Facets of Expert (e.g. power engineer) Trust in AI

- Autonomy (distributed agents)
- Beneficence (useful for all)
- Nonmaleficence (no harm)
- Justice (fairness)
- Explainability (in "power engineer" terms)
- Prepardness (for rare but possibly devastating events)

Physics Informed Machine Learning: Principal Ideas (active discussions)

- A-Priori: run a (physics-blind) ML scheme, check physics
  - Diagnostics: hierarchy of tests
- A-Posteriori: embed physics in ML
  - Loss Function
  - Graphical Model (structure=explainable) Learning
  - What we know (structure) vs what we do NOT know (NN)
- Model Reduction
  - Check Hypotheses, Phenomenologies (e.g. forgotten)
- Derive New/Old Physical Laws

Show on Applications (Examples)

- **Fluid Mechanics**
  - multi-physics
  - climate
  - astro
- Materials
- Bio-systems

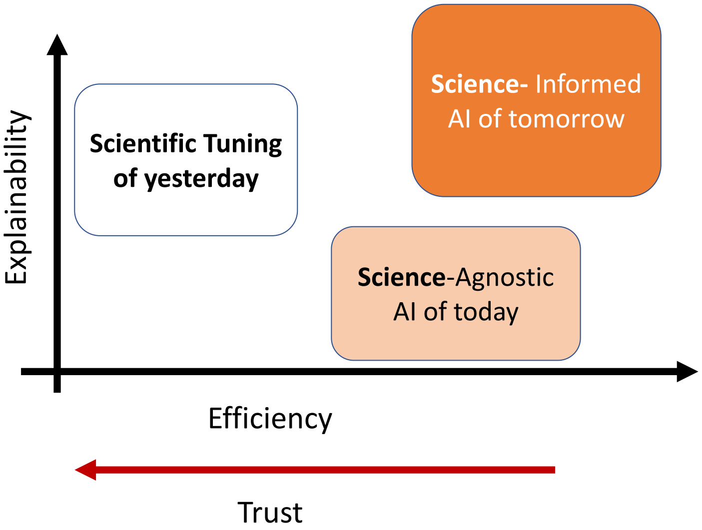
Physics & Engineering
Informed & **Reliable**

& Math-Enabled
AI & Data Science

- **Energy Systems**
  - power
  - natural gas
  - district heating/cooling
- Communication Systems
  - optical
  - quantum
- Other Systems
  - water
  - transport

- **Public Heallth**
- Epidemiology
waves of Pandemic
  - modeling
  - Mitigation
  spatio-temporal

- InferLO: Inference, Learning & Optimization
  - Graphical Models + NN
  - Control [Systems]
  - Information Theory [Error Correction]



- Fluid Mechanics
- Energy Systems
- Epidemiology
- ...

Basics of Data Science Approach [supervised, model reduction

- **Data** = Ground Truth (Resolved DNS) = (**x=input**,y=output)

- **Training**:
  = **Optimization**

$$\underset{\vartheta}{\text{argmin}}\ \textbf{Distance}(\mathbf{y}\ \&\ \textbf{Model}\ (\mathbf{x}|\boldsymbol{\vartheta} = \textbf{parameters})\ )$$

  - **Science-Agnostic Model** [e.g. **Neural Network**]
  - ...
  - ***Science-Informed Model***
    *[e.g. parameterized heuristics/**phenomenologies**]*

range(s) of modeling options

- **Efficient Execution/Inference (& Validation/tests)**

Outline

1. Introduction: Scientific AI
   - Modern (Traditional + Contemporary) Applied Math
   - Trustworthy Scientific AI
2. Trustworthy ML for Power Systems
   - Physics-Informed ML for State & Parameter Estimation
   - Which Neural Network to Choose (to localize fault)?

Trustworthy Machine Learning for Power Systems

Machine Learning (e.g. Neural Network, Graph Models, etc)

- will make Power System Computations
  - faster (efficient)
  - possible even when data/measurements incomplete
- requires ground-truth data
  - actual measurements (Phasor Measurement Units, etc)
  - power flow solvers (microscopic simulations) – reliable, possibly heavy
- can be power-system "informed" vs "agnostic"
- methods/options are many
  - should be gauged to available data, level of uncertainty, etc

Trustworthy Machine Learning for Power Systems

- **Physics-Informed** Graphical Neural Network for **Parameter & State Estimations** in Power Systems https://arxiv.org/abs/2102.06349 (with **Laurent Pagnier**)
- **Embedding Power Flow** into Machine Learning for Parameter and State Estimation https://arxiv.org/abs/2103.14251 (with **Laurent Pagnier** – not discussing today)
- **Which Neural Network to Choose** for Post-Fault Localization, Dynamic State Estimation and Optimal Measurement Placement in Power Systems? https://arxiv.org/abs/2104.03115 (with **Andrei Afonin**)

Incomplete Review: Brief, Recent, Biased

Machine Learning in Power Systems

- Structure Learning, Sparse Measurements, Graphical Models, Focus on Power Distribution: Deka, et al [2016-2019]
- Learning ODE: Power Transmission, Dynamic Coefficients in Swing Equations, Deterministic and Stochastic, Lokhov, et al [2017]
- Real-time Faulted Line Localization and PMU Placement in Power Transmission through CNN: Li, et al [2018]
- Physics Informed (Collocation Point) NN for Power Systems: Misuris, et al [2018]
- Learning a Generator Model from Terminal Bus Data: many ML schemes, tradeoffs, ranking models according to regimes, Stulov et al [2019]
- Learning from power system data stream, phasor-detective approach, Escobar et al [2019]

Machine Learning (Neural Networks) Setting

NN models: General

- $\text{NN}_{\vec{\phi}}(\vec{x}) = \vec{y}$
  - Vector, $\vec{\phi}$, of Not-Interpretable Parameters
  - Input vector: $\vec{x}$
  - Output vector: $\vec{y}$

NN models: Loss Functions

- L2 norm $\|\cdots\|$
- Probabilistic (Cross Entropy or Kullback-Leibler)
- Regularizations, e.g. L1 (sparsity, physical, etc)

NN models: Architectures

- Convolutional NN (LeCun 1989 –)
- Graph NN (Scarcelli. et al 2009 –)
- Neural ODE (Chen et al 2008 –)
- Physics Informed NN (PINN) (Raissi et al 2019 –)
- Hamiltonian NN (Greydanus et al 2018 –)

Power Flow Equations

- grid-graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- complex-valued powers: $\forall a \in \mathcal{V}: \; S_a \equiv p_a + \boldsymbol{i} q_a$
- complex-valued (electric) potentials, $\forall a \in \mathcal{V}: \; V_a \equiv v_a \exp(\boldsymbol{i}\theta_a)$,
- Power Flow (PF) equations:

$$p_a = \sum_{b;\{a,b\}\in\mathcal{E}} v_a v_b \Big[g_{ab} \cos\left(\theta_a - \theta_b\right) + \beta_{ab} \sin\left(\theta_a - \theta_b\right) \Big],$$
$$q_a = \sum_{b;\{a,b\}\in\mathcal{E}} v_a v_b \Big[g_{ab} \sin\left(\theta_a - \theta_b\right) - \beta_{ab} \cos\left(\theta_a - \theta_b\right) \Big],$$

- Direct PF Map: $\boldsymbol{\Pi}_{\boldsymbol{Y}} : \boldsymbol{S} \equiv (S_a | a \in \mathcal{V}) \mapsto \boldsymbol{V} \equiv (V_a | a \in \mathcal{V})$ - implicit (need to solve eqs. - not discussed today `https://arxiv.org/abs/2103.14251`)
- Inverse PF Map: $\boldsymbol{S} = \boldsymbol{\Pi}_{\boldsymbol{Y}}^{-1}(\boldsymbol{V})$ – explicit (do not need to solve eqs.)

Task #1: State & Parameter Estimation

- State Estimation
  - Full Observability: given $\mathcal{G}$ and $\boldsymbol{Y}$ to estimate injected/consumed active and reactive powers = application of the inverse PF map, $\mathbf{\Pi}^{-1}$
  - Limited Observability:
    - Complement Missing power injections/consumptions at the nodes where voltages and phases are measured
    - Challenging Version: to reconstruct injected/consumed powers and also voltages and phases at all nodes of the system. (super-resolution – will not discuss)
- Parameter Estimation
  - Reconstruct Graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and line characteristics, $\boldsymbol{Y}$

Task #1: SE & PE. Reduced Modeling.

- Setting of Partial Observability
- Find Equivalent (Reduced) Model of Power System
- "Inspired" by Kron Reduction
  - $\boldsymbol{I}^{(o)} = \boldsymbol{Y}^{(r)} \boldsymbol{V}^{(o)}$
  - "o" - observed; "r" - reduced
  - $\mathcal{G}^{(r)} \equiv (\mathcal{V}^{(o)}, \mathcal{E}^{(r)})$
  - $\boldsymbol{Y}^{(r)} \doteq (\{a,b\} | Y_{ab}^{(r)} \neq 0)$ – associated with the effective (not necessarily real) power lines, $\{a,b\} \in \mathcal{E}^{(r)}$. $\boldsymbol{Y}^{(r)}$
- Reduced Model
  - $\boldsymbol{S}^{(o)} = \Pi^{-1}_{\boldsymbol{Y}^{(r)}}(\boldsymbol{V}^{(o)})$
  - Learn it !?

Task #1: SE & PE. PIML of Reduced Model

- Power Graphical NN:

$$\min_{\varphi, \boldsymbol{Y}^{(r)}} L_{\text{Power-GNN}}\left(\varphi, \boldsymbol{Y}^{(r)}\right),$$

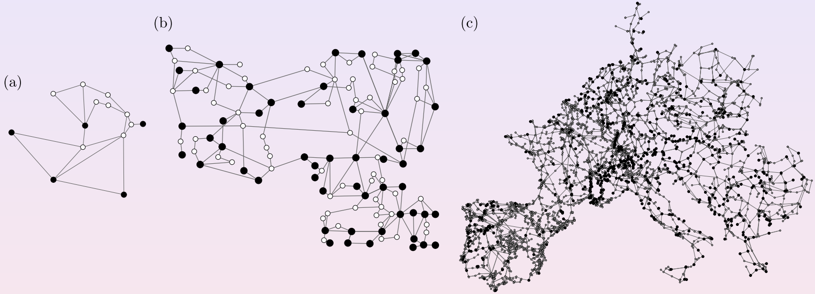
$$L_{\text{Power-GNN}}\left(\varphi, \boldsymbol{Y}^{(r)}\right) \equiv \frac{1}{N|\mathcal{V}^{(o)}|} \sum_{n=1}^{N} \left\| \boldsymbol{S}_n^{(o)} - \underbrace{\Pi^{-1}_{\boldsymbol{Y}^{(r)}}\left(\boldsymbol{V}_n^{(o)}\right)}_{\text{physics = interpretable}} - \underbrace{\Sigma_{\varphi}\left(\mathcal{V}_n^{(o)}, S_n^{(o)}\right)}_{\text{NN = "sub-scale"}} \right\|^2 + \underbrace{\mathcal{R}(\varphi)}_{\text{regularization}}$$

- SIMULTANEOUSLY physics-informed and physics-blind parts
- Compare with Vanilla-NN

$$L_{\text{NN}} \doteq \frac{1}{N|\mathcal{V}^{(\mathbf{o})}|} \sum_{n=1}^{N} \left\| \boldsymbol{S}_n^{(o)} - \text{NN}_{\varphi}\left(\boldsymbol{V}_n^{(o)}\right) \right\|^2$$

Task #1: SE & PE. Power GNN vs Vanilla NN. Experiments

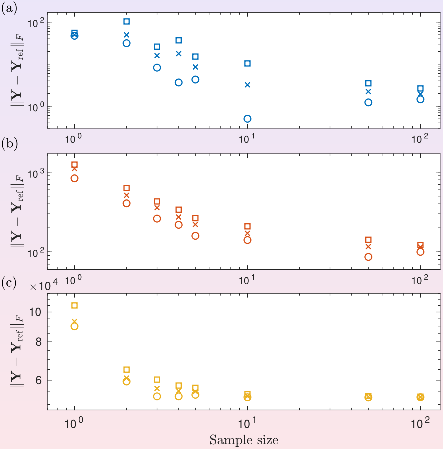
IEEE 14-bus [panel (a)], IEEE 118-bus [panel (b)] and PanTaGruEl [panel (c)] models



State Estimation Test: Six set of samples were generated for each network. Average mismatch of predicted power injections (on the training set in parentesis)

	case #1	case #2	case #3	case #4	case #5	case #6
Vanilla NN	4.9E-6	7.2E-5	6.3E-3	5.2E-2	6.3E-2	1.4E0
	(4.2E-6)	(6.6E-5)	(5.0E-5)	(3.7E-5)	(1.2E-4)	(4.2E-6)
Power-GNN	3.0E-6	5.8E-7	6.9E-7	1.3E-6	2.9E-7	3.0E-6

Task #1: SE & PE. Power GNN vs Vanilla NN. Experiments



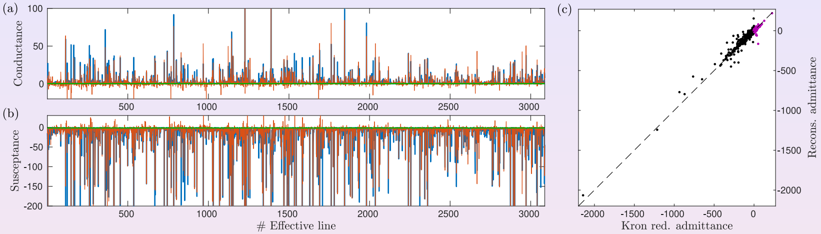
Full Observability. Parameter Estimation.

- Reconstruction of the admittance matrix $\boldsymbol{Y}$ for IEEE 14-bus (a), IEEE 118-bus (b) and PanTaGruEl (c) models
- The min, mean and max values are displayed as circles, crosses and squares respectively (for 10 realizations.)

Notice !!

- Quality of the reconstruction by Power-GNN – especially for large network

Task #1: SE & PE. Power GNN vs Vanilla NN. Experiments



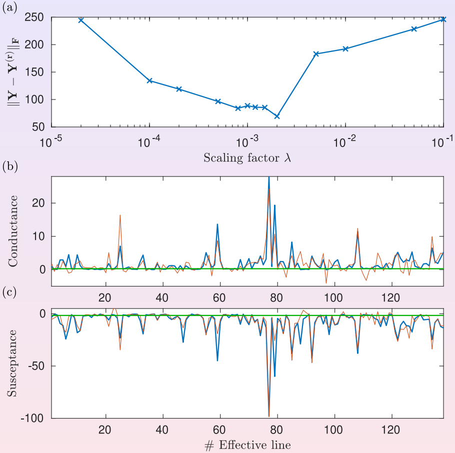
Partial Observability. Parameter Estimation. PanTaGruEl model

- Initial (pre-training) values – in green.
- Trained values and their Kron-reduction counterparts – red and blue respectively.
- (c) shows alternative visualization of the reference-vs-predicted values of the line conductances (purple) and susceptances (black)

Notice !!

- Quality of the reconstruction by Power-GNN – especially for large network

Task #1: SE & PE. Power GNN vs Vanilla NN. Experiments



Partial Observability. Parameter Estimation. Medium model. Dependence of the quality of the PE reconstruction on regularization

- (a) Dependence of the integrated (over all the nodes) quality indicator on the regularization parameter
- (b) and (c) shows, respectively, conductances and susceptances of the PS lines reconstructed at the optimal value of the regularization parameter, correspondent to the minimum of the curve shown in panel (a).
- (b) and (c) also shows, for reference, respective values the Kron-reduction counter-parts.

Notice !!

- Corrections due to NN (sub-scale contributions) may be significant!

Power Flow Equations (A bit more general setting. Static.)

$\text{PF}_{\vec{\psi}}(\vec{x}) = 0$

- vector of "fluctuating" variables (samples): $\vec{x} = \vec{S} \cup \vec{V}$
- sufficient number of samples: $n = 1, \cdots, N: \quad \vec{x}_n$
  - Complex (apparent) powers across the grid: $\vec{S} = (S_a = P_a + iQ_a | a \in \mathcal{V})$
  - Complex voltages (potentials) across the grid: $\vec{V} = (V_a | a \in \mathcal{V})$
- $\vec{\psi}$ - vector of interpretable "quenched" (frozen) parameters
  - The grid=graph: $\mathcal{G} \doteq (\mathcal{V}, \mathcal{E})$
  - Matrix of admittance: $\hat{Y} = (y_{ab} | \{a, b\} \in \mathcal{E})$

- $\forall a \in \mathcal{G}: S_a = V_a \sum_{\{a,b\} \in \mathcal{E}} \left(\frac{V_a - V_b}{z_{ab}}\right)^*$

Power Flow Dynamics = Differential Algebraic Equations

$\text{PF-DAE}_{\vec{\psi}}(\{\vec{x}(t)\}) = 0$

- vector of "fluctuating" trajectories (samples): $\{\vec{x}(t)\} = \{\vec{S}(t)\} \cup \{\vec{V}(t)\}$
- sufficient number of samples: $i = 1, \cdots \quad \{\vec{x}_i(t)\}$
  - Complex (apparent) powers across the grid evolving in time: $\{\vec{S}(t)\} = \{S_a(t) = P_a(t) + iQ_a(t) | \forall t;\ a \in \mathcal{V}\}$
  - Complex voltages (potentials) across the grid evolving in time: $\{\vec{V}(t)\} = \{V_a(t) = v_a e^{i\theta_a} | \forall t,\ a \in \mathcal{V}\}$
- $\vec{\psi}$ - vector of interpretable "quenched" (frozen) parameters
  - The grid=graph: $\mathcal{G} \doteq (\mathcal{V}, \mathcal{E})$
  - Matrix of admittance: $\hat{Y} = (y_{ab} | \{a, b\} \in \mathcal{E})$

$$\forall a \in \mathcal{G}: \quad 0 = \text{Im}\left(S_a - V_a \sum_{\{a,b\}\in\mathcal{E}} \left(\frac{V_a - V_b}{z_{ab}}\right)^*\right)$$

$$M_a \ddot{\theta}_a + \beta_a \dot{\theta}_a = \text{Re}\left(S_a - V_a \sum_{\{a,b\}\in\mathcal{E}} \left(\frac{V_a - V_b}{z_{ab}}\right)^*\right)$$

- AC-Swing (DAE) Eqs.

Task #2: Failure Detection – Localize Failure

- $\vec{\psi} = (\underbrace{\vec{\psi}^{(b)}}_{\text{before}}, \underbrace{\vec{\psi}^{(a)}}_{\text{after}})$
- $\vec{\phi}$ - not physical (NN) parameters

- $\text{NN}^{(\text{FD})}_{\vec{\phi}}(\vec{x}_o^{(b)}, \vec{x}_o^{(a)}) = (\vec{x}_u^{(a)}, \vec{\psi}^{(a)})$
- $\text{PF}_{\vec{\psi}^{(a)}}\left(\vec{x}_o^{(a)}; \vec{x}_u^{(a)}\right) = 0$
- $\text{NN}^{(\text{SR})}_{\vec{\phi}}(\vec{x}_o|\psi) = \vec{x}_u$ - unobserved (Super Resolution)

Power Flow Agnostic Machine Learning

- $\min\limits_{\vec{\phi}} \sum\limits_{n=\text{samples}} \|\text{NN}^{(\text{FD})}_{\phi}(\vec{x}_o^{(n;b)}, \vec{x}_o^{(n;a)}) - (\vec{x}_u^{(n;a)}, \vec{\psi}^{(n;a)})\|$

Power Flow Informed Machine Learning

- $\min\limits_{\vec{\psi},\vec{\phi}} \sum\limits_{n} \left|\text{PF}_{\vec{\psi}}\left(\vec{x}_o^{(n)}; \text{NN}^{(SR)}_{\vec{\phi}}(\vec{x}_o^{(n)}|\psi)\right)\right|$ (only after)
- may also do dynamic version (not shown)

Task #2: Failure Detection – Localize Failure

Previous Work (Static Measurements):

- Real-time Faulted Line Localization and PMU Placement in Power Transmission through CNN: Li, Deka, Wang, MC [2018]

4640 IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 34, NO. 6, NOVEMBER 2019

Real-Time Faulted Line Localization and PMU Placement in Power Systems Through Convolutional Neural Networks

Wenting Li, *Student Member, IEEE*, Deepjyoti Deka, *Member, IEEE*, Michael Chertkov, *Senior Member, IEEE*, and Meng Wang, *Member, IEEE*

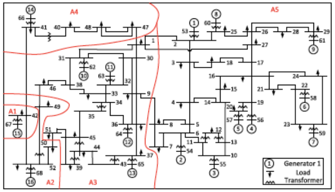


Fig. 1. IEEE 68-bus system with five coherence groups [19].

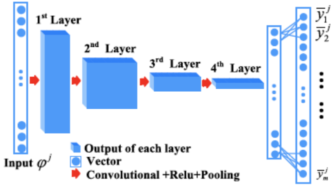
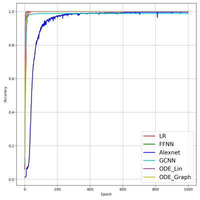


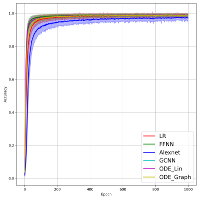
Fig. 3. The structure of the proposed CNN.

Task #2: Failure Detection – Localize Failure

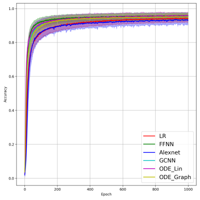
Static Measurements: Partial Observability



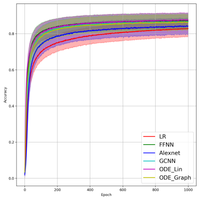
(a) 100%



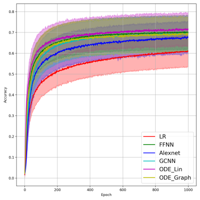
(b) 70%



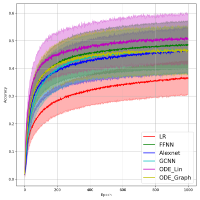
(c) 40%



(d) 20%



(e) 10%



(f) 5%

Task #2: Failure Detection – Localize Failure

Static Measurements:

- Loss Function (physics-agnostic) - Cross Entropy, Adam (gradient descent) optimizer, trained for 1000 epochs
- Compared Linear Regression, Feed-Forward NN, Graph-CNN, Neural-ODE
- Accuracy (measure of performance) = proportion of correct predictions

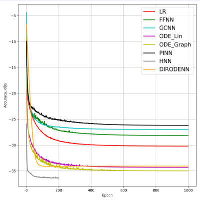
		Model comparing	
Model	Performance*	Num. of params	CPU[1] Ti
LR	0.4993	6,003	0.016
FFNN	0.6490	5,079	0.021
AlexNet [?]	0.6229	2,071	0.100
GCNN	0.6342	5,079	0.029
ODE Lin	0.6737	10,695	1.238
ODE Graph	0.6398	10,695	1.284

- *5 % observability, 0 % SNR
- **Time for epoch averaged over 1000 epochs

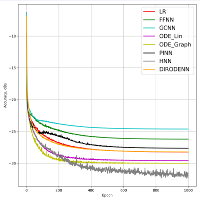
[1] Intel(R) Xeon(R) CPU @ 2.20GHz
[2] 12GB NVIDIA Tesla K80 GPU

Task #2: Failure Detection – Localize Failure

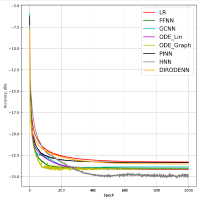
Dynamic Measurements: Partial Observability



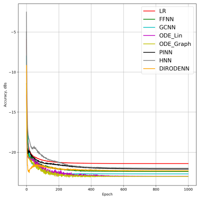
(g) 100%



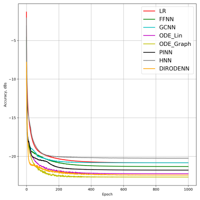
(h) 70%



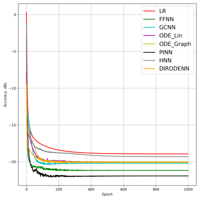
(i) 40%



(j) 20%



(k) 10%



(l) 5%

Task #2: Failure Detection – Localize Failure

Dynamic Measurements:

- Loss Function - Mean-Square-Error, Adam (gradient descent) optimizer, trained for 1000 epochs (with some variations)
- Compared LL, FFNN, Graph-CNN, Neural ODE-Lin, Neural ODE-Graph, PINN, Hamiltonian NN, Neural Swing-Eq-ODE
- Accuracy [metric of performance] = Loss in dBs (Normalized Mean-Square-Error)

Task #3: Optimal Placement of Measurements

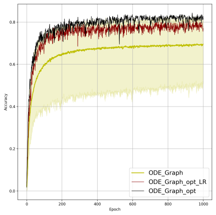
NN for PMU Placement:

- Better Failure Detection
- Reinforcement Learning of a kind

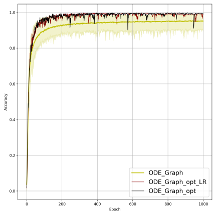
- Built on a Failure Detection Algorithm (FDA) – any from #2
- Use the FDA to generate k-instances, $f: \vec{x} \to \vec{y}$
  - $\forall k: \ \vec{x}_k$ – vector of PMU placement (fixed cardinality)
  - $\forall k: \ \vec{y}_k$ – score (repetitive use of the FDA Alg.)

- Train $NN^{(OP)}: \vec{x} \to \vec{y}$ over k-instances (ground truth)
  - $\vec{\phi}_* = \arg\min_{\vec{\phi}} \sum_k \| NN^{(OP)}_{\vec{\phi}}(\vec{x}_k) - \vec{y}_k \|$
- Maximize score (adaptively): $\max_{\vec{x}} NN^{(OP)}_{\vec{\phi}_*}(\vec{x})$

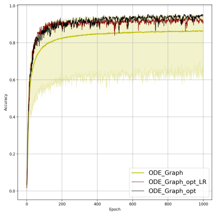
Task #3: Optimal Placement of Measurements



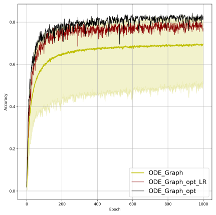
(m) Graph ODE: 70%



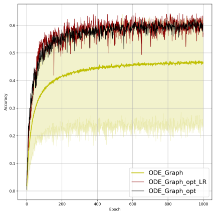
(n) Graph ODE: 40%



(o) Graph ODE: 20%

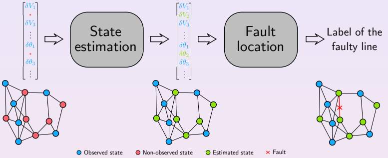


(p) Graph ODE: 10%



(q) Graph ODE: 5%

Working on



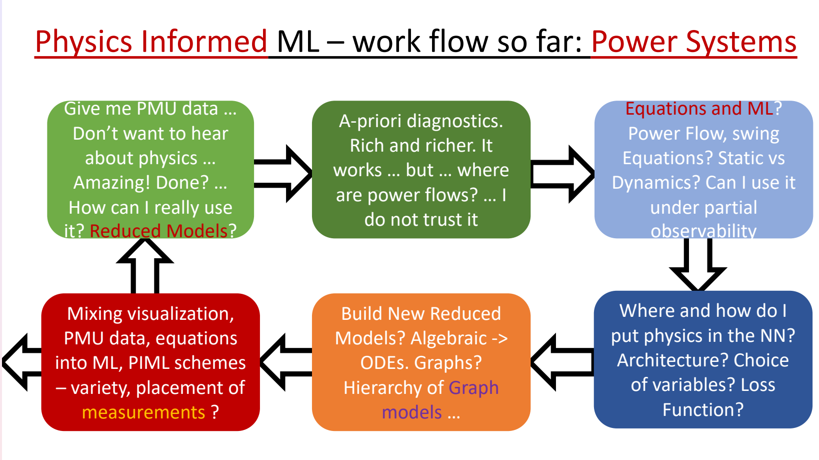
State & Parameter Estimation ⇒ Fault Localization

- scaling it up to larger systems
  - embedding more physics
- meta-learning (train on small, use on large)

Working on

Spatio-temporal coarse-graining – model reduction

- "Structure- & Physics- Preserving Reductions of Power Grid Models", Colin Grudzien, Deepjyoti Deka, Michael Chertkov, Scott N Backhaus https://arxiv.org/abs/1707.03672
  - ⇒ better (faster & more accurate) **Graph-reduction**
- "Model Reduction of Swing Equations with Physics Informed PDE" Laurent Pagnier, Michael Chertkov, Julian Fritzsch, Philippe Jacquod https://arxiv.org/abs/2110.14066
  - ⇒ **PDE Learning**


Physics Informed ML – work flow so far: Power Systems
Give me PMU data … Don't want to hear about physics … Amazing! Done? … How can I really use it? Reduced Models?
A-priori diagnostics. Rich and richer. It works … but … where are power flows? … I do not trust it
Equations and ML? Power Flow, swing Equations? Static vs Dynamics? Can I use it under partial observability
Where and how do I put physics in the NN? Architecture? Choice of variables? Loss Function?
Build New Reduced Models? Algebraic -> ODEs. Graphs? Hierarchy of Graph models …
Mixing visualization, PMU data, equations into ML, PIML schemes – variety, placement of measurements ?

Andrey Afonin

Chris Koh

Laurent Pagnier

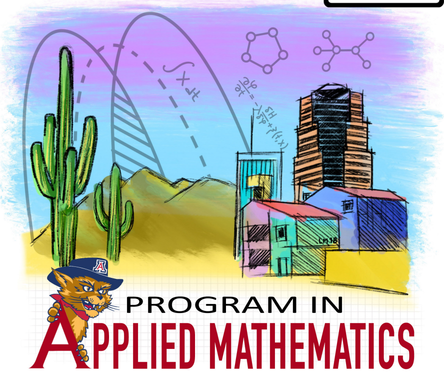
Nick Stulov

- Research focused, since 1976, one of the US first [dynamical systems, integrability, turbulence …]
- Interdisciplinary: 100+ professors/ 26 departments/ 8 colleges across UA campus (CoS & CoE & Optics – top 3)
- Mixing traditional @ contemporary Applied Math
- Graduate, Ph.D. focused, no terminal M.Sc.
- 60 Ph.D students (**13**/16/10 enrolled in **2021**/20/19)
- **3 Core Courses** (1st year -- Methods, Analysis, Algorithms) https://appliedmath.arizona.edu/students/new-core-courses
- Strong collaborations with Industry (e.g. Raytheon, Uber, Intel, Critical Path, etc) and National Labs (e.g. LANL, LLNL, NREL, NNSS, etc)
- 5 seminar/colloquium series – recorded and posted online
- Participation in many UA & National Edu Projects

http://appliedmath.arizona.edu/

chertkov@arizona.edu

SCAN ME



Support is Appreciated !!

- **Energy Systems**: UArizona start up + DOE/ARPA-E

Thanks for your attention !