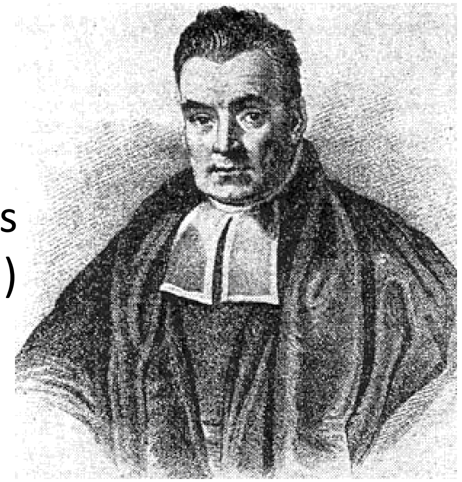
Data-Driven Optimization using **Limited Data:** the power of



Thomas Bayes
(with high probability, this is him)

Parikshit Pareek

Sidhant Misra

Kaarthik Sundar

Los Alamos
NATIONAL LABORATORY

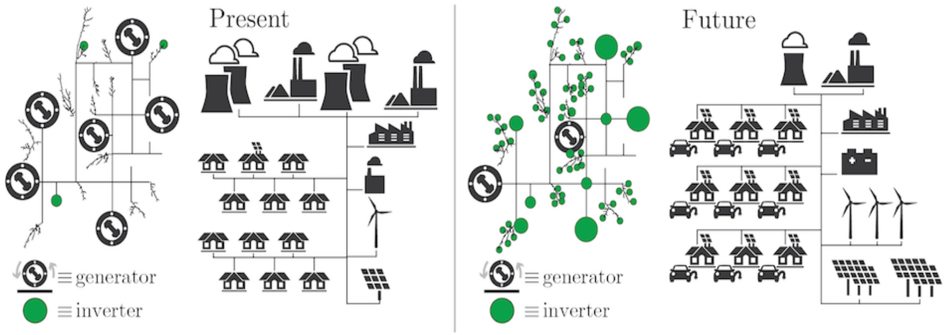
Deep Deka
Research Scientist,

MITei
MIT Energy Initiative

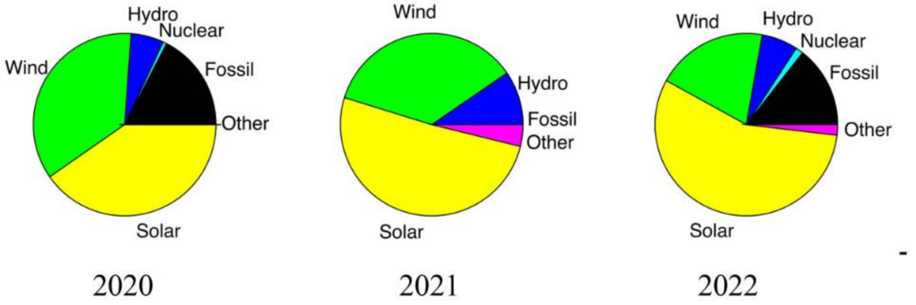
Energy Infrastructure of the future: the case for **resilience**

- Renewables and smart devices are leading to a paradigm shift in grid operations.
  - Greater Variability/intermittent
  - Lesser inertia/stability
  - More measurements and data-driven capabilities

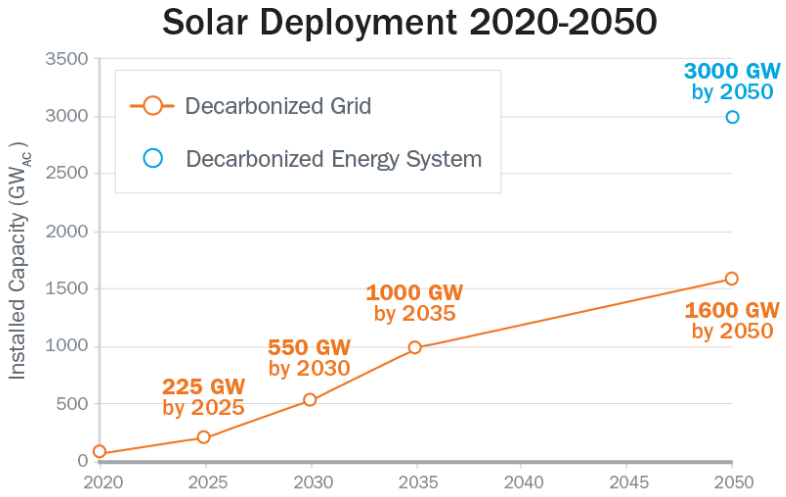
- Need: faster but **risk-aware** data-driven decision making



Smart but *stochastic*

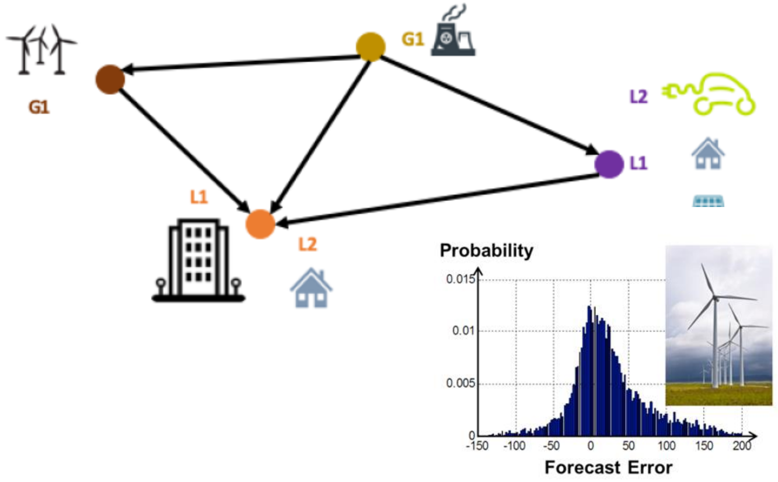


Global new generation



Transmission Grid Optimization under uncertainty

Optimal Power Flow: minimize generation costs while satisfying injection/demand, technical constraints



Problem:

$$\min \sum_{g\in\mathcal{G}} c(p_g) \qquad \text{Cost}$$

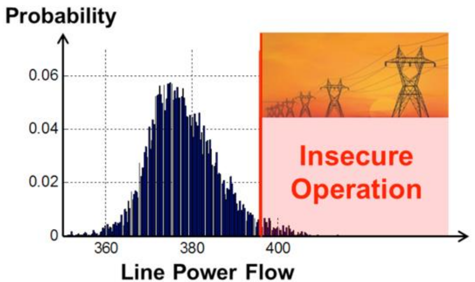
$$\text{s.t.} \quad \sum_{g\in\mathcal{G}_i} p_g - P_i = \sum_{j\in\mathcal{B}} v_i v_j (G_{ij}\cos(\theta_i-\theta_j) + B_{ij}\sin(\theta_i-\theta_j)) \quad \forall i \in \mathcal{B}$$

$$\sum_{g\in\mathcal{G}_i} q_g - Q_i = \sum_{j\in\mathcal{B}} v_i v_j (G_{ij}\sin(\theta_i-\theta_j) - B_{ij}\cos(\theta_i-\theta_j)) \quad \forall i \in \mathcal{B}$$

$$(p,q,v,\theta) \in \mathcal{T}$$

Non-linear power flow physics

Safety Constraint



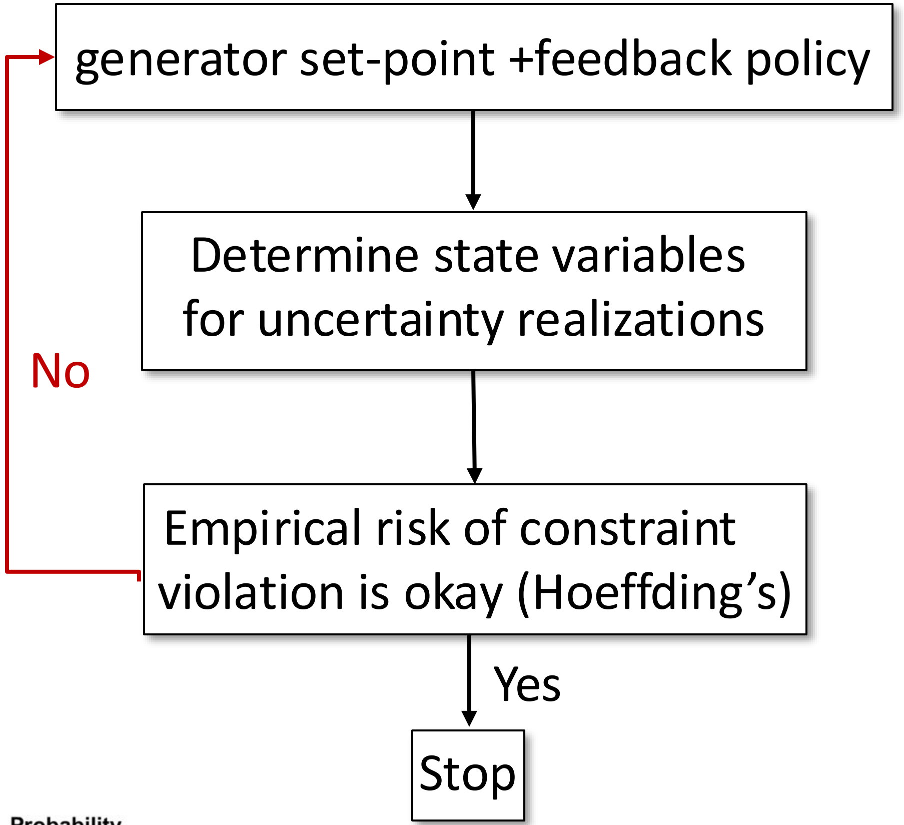
Hard to Solve generally:

- ✓ *Approximate analytical optimization* (*linear models,* Gaussian uncertainty)
- ✓ *ML surrogates* ***+ validation***

ML based: Jalali, Pareek, Velloso, Zamzam, Singh, Kekatos, Baker, Bernstein, van- Hentenryck, Fioretto, Donti, Chatzivasileiadis, Misra, Nagarajan, Zhu, Qiu ….(incomplete)

Work-Flow for reliable grid decisions

Every 5- 15 mins:



generator set-point +feedback policy

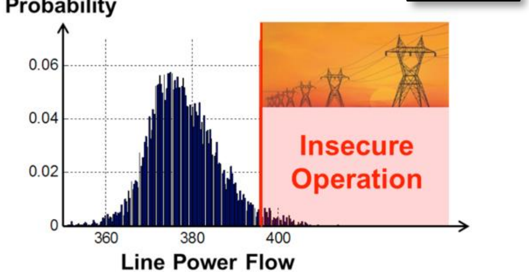
Determine state variables for uncertainty realizations

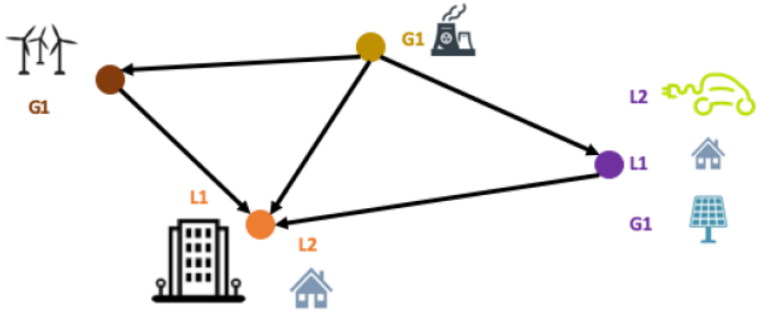
Empirical risk of constraint violation is okay (Hoeffding’s)

No

Yes

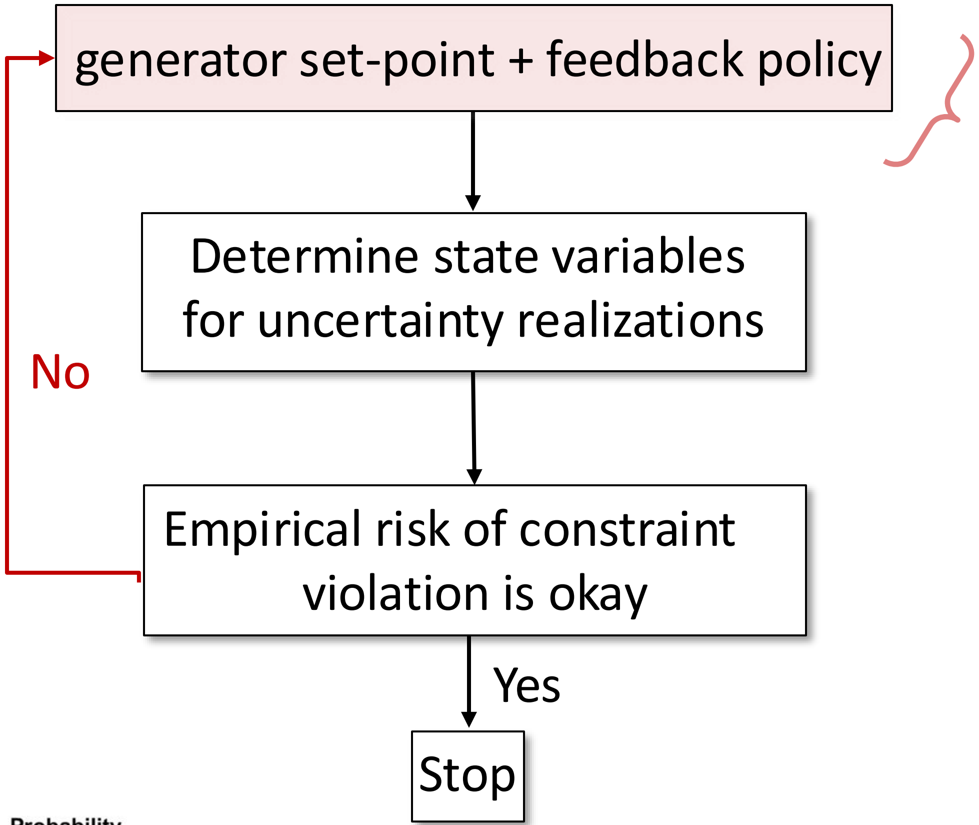
Stop





Improvements in the Work-Flow

Every 5- 15 mins:



generator set-point + feedback policy

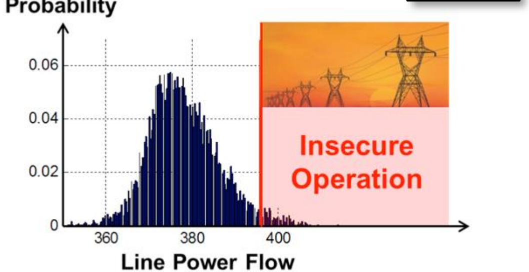
Determine state variables for uncertainty realizations

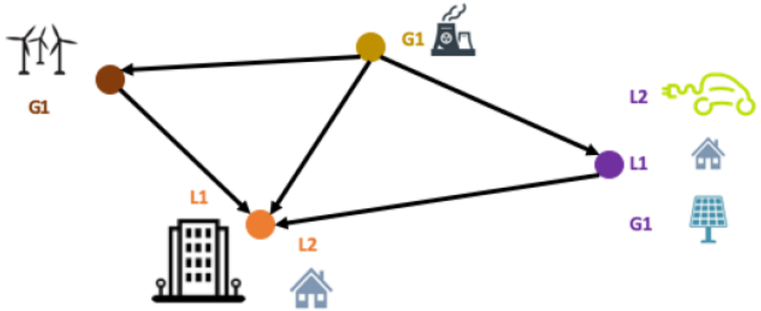
Empirical risk of constraint violation is okay

No

Yes

Stop

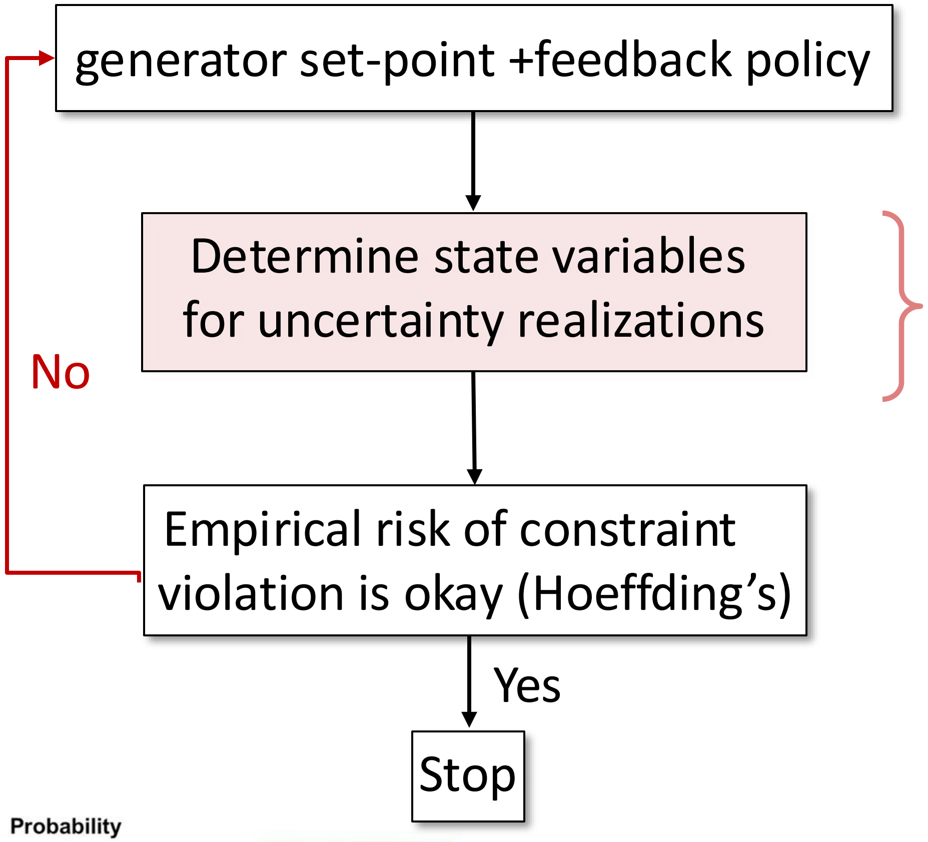




- Optimization with Hard non-convex/ non-linear constraints
- Needs to be solved fast
- Limited training data if system changes

Improvements in the Work-Flow

Every 5- 15 mins:



generator set-point +feedback policy

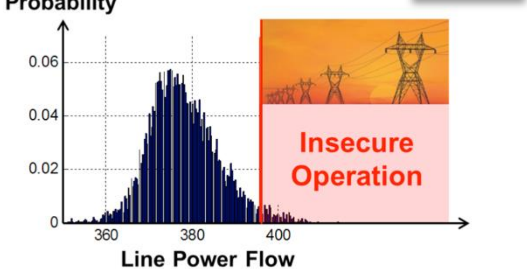
Determine state variables for uncertainty realizations

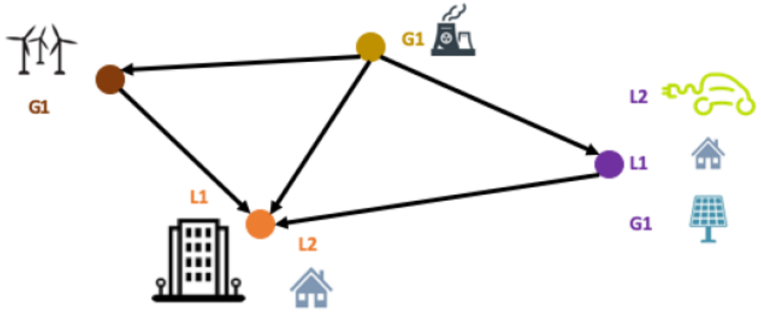
Empirical risk of constraint violation is okay (Hoeffding’s)

No

Yes

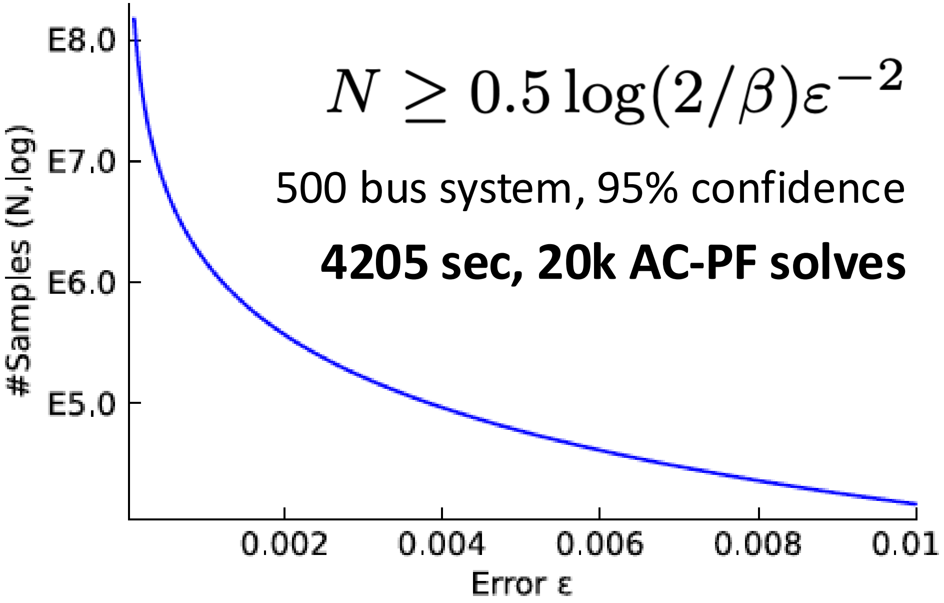
Stop





- VOLTAGE is **implicit, non-linear** function of Injection

$$\overset{\text{Net Power Injection}}{S_i} = \sum_{j\in\mathcal{N}} \underset{\text{Network Parameter}}{Y^{\dagger}_{ij}}(v_i v_i^{\dagger} - \overset{\text{Complex Voltage}}{v_i} v_j^{\dagger})$$

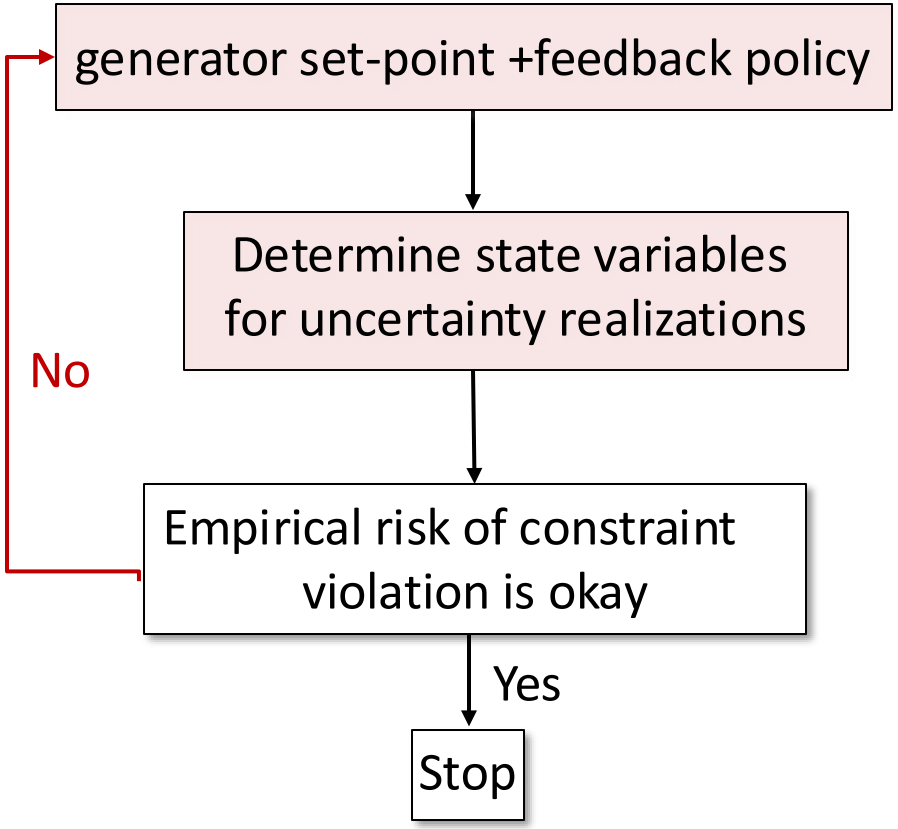


$$N \geq 0.5\log(2/\beta)\varepsilon^{-2}$$

500 bus system, 95% confidence

4205 sec, 20k AC-PF solves

Every 5- 15 mins:



generator set-point +feedback policy

Determine state variables for uncertainty realizations

Empirical risk of constraint violation is okay

No

Yes

Stop

Research Question:

a. Can we design ML optimization models that use limited data for solutions with confidence?

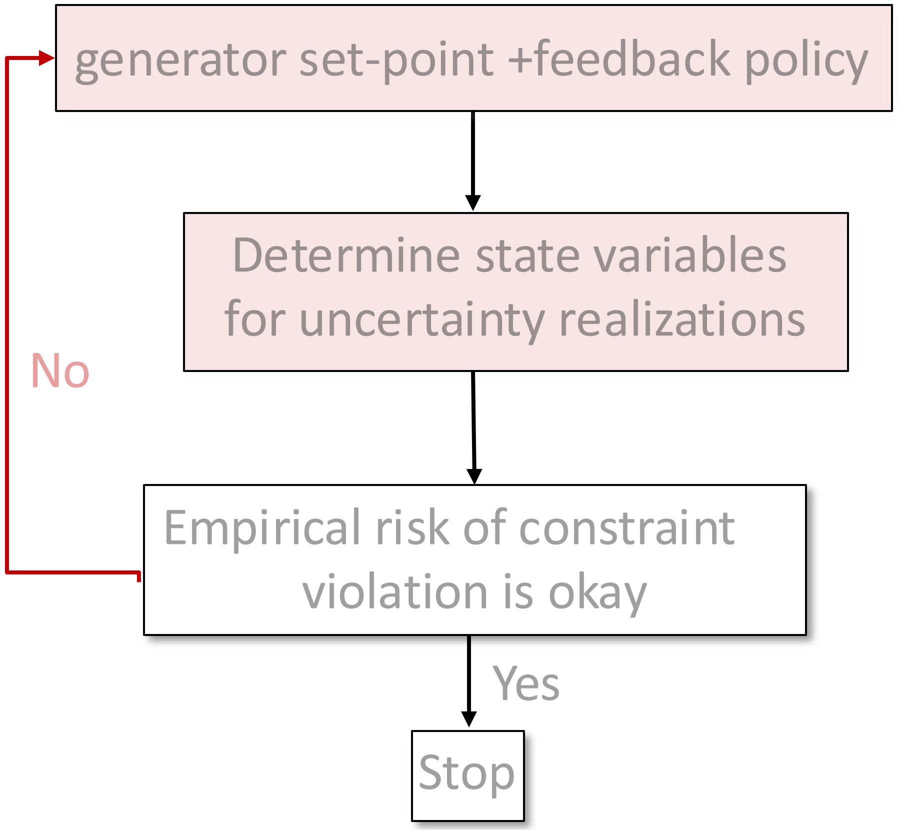
b. Can we design injection S → voltage V map for faster risk assessment using limited data?

Solution: **Bayesian** machine learning

a. Semi-supervised Bayesian Neural network for OPF

b. Network-aware Gaussian Process for voltage maps

Every 5- 15 mins:


generator set-point +feedback policy

Determine state variables for uncertainty realizations

Empirical risk of constraint violation is okay

No

Yes

Stop

Parameters: w, input: x, output: y

Research Question:

a. Can we design ML optimization models that use limited data for solutions with confidence?

b. Can we design injection S → voltage V map for faster risk assessment using limited data?

Solution: **Bayesian** machine learning

a. Semi-supervised Bayesian Neural network for OPF

b. Network-aware Gaussian Process for voltage maps

- Evaluate Posterior of parameters given prior and data

$$p(w|\mathbf{x},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x},w)\,p(w)$$

- Estimate Posterior prediction of output for new input

$$p(\mathbf{y}^t|\mathbf{x}^t,\mathcal{D}) = \mathbb{E}_{p(w|\mathcal{D})}[p(f_w(\mathbf{x}^t)]$$

b. Can we design a data-driven input-output (injection S → voltage V) map?

$$\overset{\text{Net Power Injection}}{S_i} = \sum_{j\in\mathcal{N}} \underset{\text{Network Parameter}}{Y^{\dagger}_{ij}}\,(v_i v_i^{\dagger} - \overset{\text{Complex Voltage}}{v_i}\, v_j^{\dagger})$$

Properties of a `**good’** approximator:

- Explicit S → V
- Easy to Evaluate, Differentiable
- Interpretable in terms of network structure
- Re-trainable/ transferable

Gaussian Process Regression for injection S → voltage V

$$\overset{\text{Net Power Injection}}{S_i} = \sum_{j \in \mathcal{N}} \underset{\text{Network Parameter}}{Y^{\dagger}_{ij}} (v_i v_i^{\dagger} - \overset{\text{Complex Voltage}}{v_i} v_j^{\dagger})$$

Properties of a `**good’** approximator:

- Explicit S → V
- Easy to Evaluate, Differentiable
- Interpretable in terms of network structure
- Re-trainable/ transferable

- Non-parametric model for V as function of injection s

$$\widehat{V} = f(\mathbf{s}) + \epsilon$$

$$f(\mathbf{s}^i) \sim \mathcal{GP}(\underset{\text{Zero Mean}}{0}, \underset{\text{Kernel Function}}{k}(\mathbf{s}^i, \mathbf{s}^j))$$

$$k(\mathbf{s}^i, \mathbf{s}^j) = \tau^2 \exp\left\{ \frac{-\|\mathbf{s}^i - \mathbf{s}^j\|^2}{2\ell^2} \right\}$$

Squared Exponential Kernel

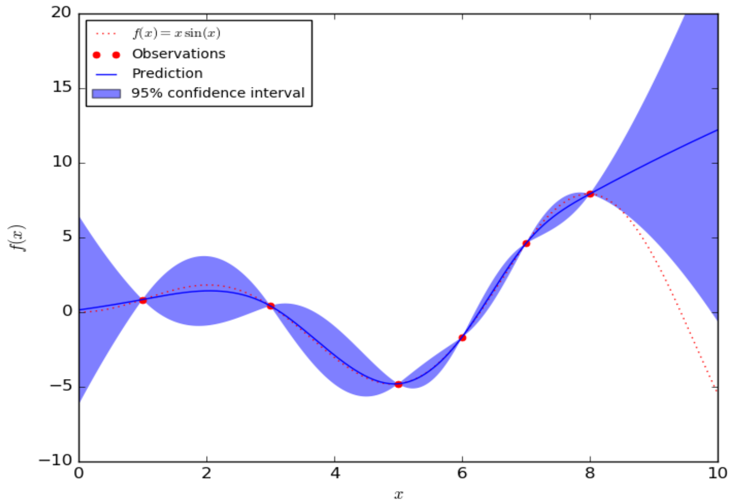
Gaussian Process Regression for injection S → voltage V

Net Power Injection

Complex Voltage

$$S_i = \sum_{j\in\mathcal{N}} Y_{ij}^{\dagger}(v_i v_i^{\dagger} - v_i v_j^{\dagger})$$

Network Parameter



- Non-parametric model for V as function of S

$$\widehat{V} = f(\mathbf{s}) + \epsilon$$

$$f(\mathbf{s}^i) \sim \mathcal{GP}(0, k(\mathbf{s}^i, \mathbf{s}^j))$$

Zero Mean

Kernel Function

$$k(\mathbf{s}^i, \mathbf{s}^j) = \tau^2 \exp\left\{\frac{-\|\mathbf{s}^i - \mathbf{s}^j\|^2}{2\ell^2}\right\}$$

Squared Exponential Kernel

- Learn the Kernel parameters using Maximum Likelihood on training data

$$S = [\mathbf{s}^1 \ldots \mathbf{s}^i \ldots \mathbf{s}^N] \qquad \widehat{\mathbf{V}}_j = [V_j^1 \ldots V_j^N]$$

- Prediction for new injection sample

$$\text{Mean}: \mathbb{E}[f(\mathbf{s})] = V_j(\mathbf{s}) = \mathbf{k}^T[K + \sigma_\epsilon^2 I]^{-1}\widehat{\mathbf{V}}_j$$

$$\text{Variance}: \sigma^2[f(\mathbf{s})] = k(\mathbf{s}, \mathbf{s}) - \mathbf{k}^T[K + \sigma_\epsilon^2 I]^{-1}\mathbf{k}$$

Network scale (Full) Gaussian Process is slow

- Prediction

$$\widehat{V} = f(\mathbf{s}) + \epsilon$$

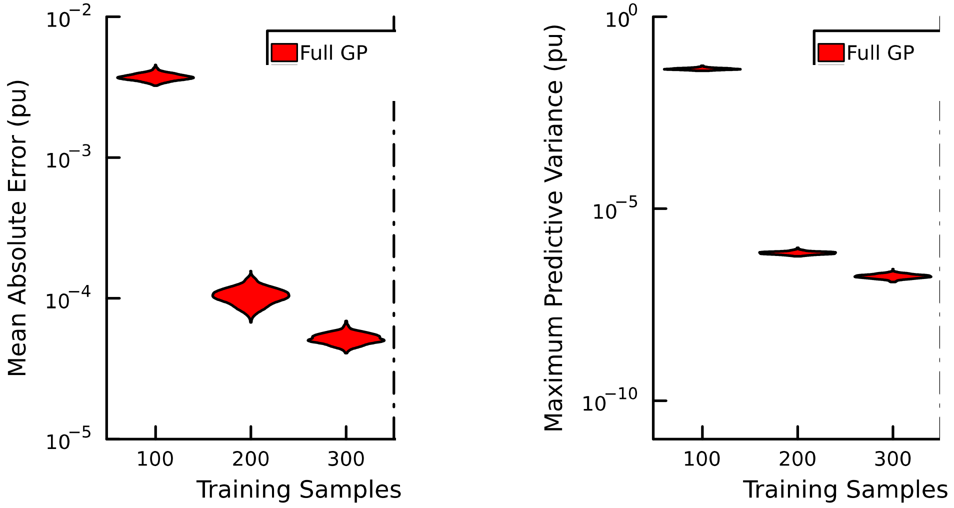
$$\text{Mean}: \mathbb{E}[f(\mathbf{s})] = V_j(\mathbf{s}) = \mathbf{k}^T[K + \sigma_\epsilon^2 I]^{-1}\widehat{\mathbf{V}}_j$$

$$\text{Variance}: \sigma^2[f(\mathbf{s})] = k(\mathbf{s}, \mathbf{s}) - \mathbf{k}^T[K + \sigma_\epsilon^2 I]^{-1}\mathbf{k}$$

Squared Exponential Kernel

$$k(\mathbf{s}^i, \mathbf{s}^j) = \tau^2 \exp\left\{\frac{-\|\mathbf{s}^i - \mathbf{s}^j\|^2}{2\ell^2}\right\}$$

- No network dependance
- Scales as *O(N^3)* with samples
- **Can we improve further?**



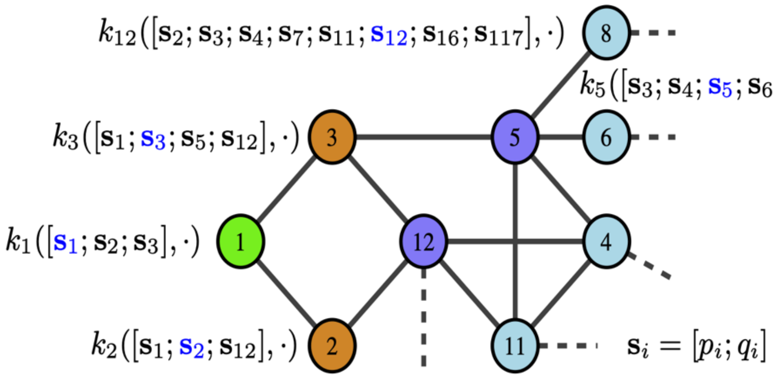
500 random trials for 500-node network
Testing 1000 out-of-sample data points.

Vertex-Degree Kernel (VDK) GP

- Additive Kernels over node-neighborhoods

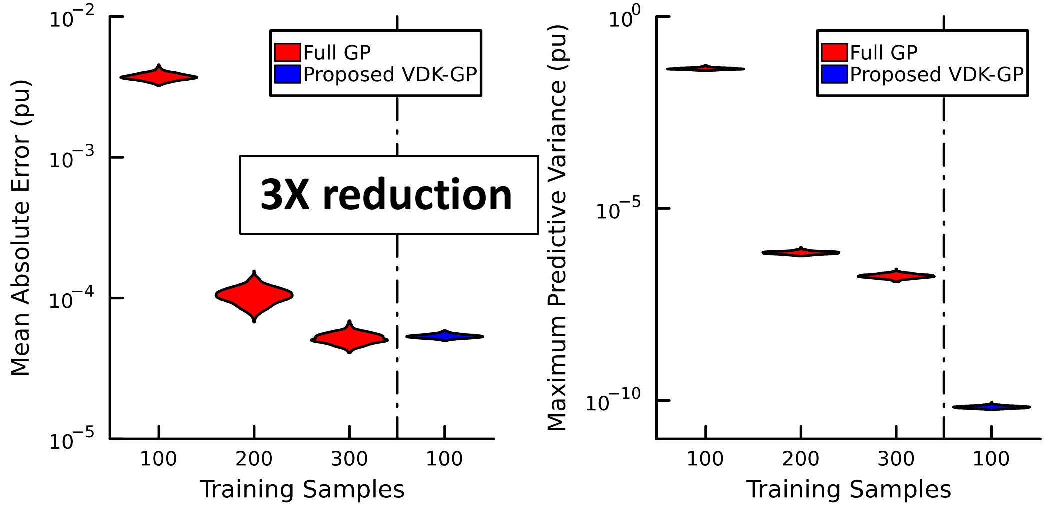
$$k_v(\mathbf{s}^i, \mathbf{s}^j) = \sum_{b=1}^{|\mathcal{B}|} k_b(\mathbf{x}_b^i, \mathbf{x}_b^j)$$

- Why?
  - Neighboring injections have correlated effect on voltage
  - Effect of far away injections is approximately independent
- Dimension Reduction
  - Effective input dimension is max. vertex degree



$k_{12}([\mathbf{s}_2; \mathbf{s}_3; \mathbf{s}_4; \mathbf{s}_7; \mathbf{s}_{11}; \mathbf{s}_{12}; \mathbf{s}_{16}; \mathbf{s}_{117}], \cdot)$
$k_5([\mathbf{s}_3; \mathbf{s}_4; \mathbf{s}_5; \mathbf{s}_6$
$k_3([\mathbf{s}_1; \mathbf{s}_3; \mathbf{s}_5; \mathbf{s}_{12}], \cdot)$
$k_1([\mathbf{s}_1; \mathbf{s}_2; \mathbf{s}_3], \cdot)$
$k_2([\mathbf{s}_1; \mathbf{s}_2; \mathbf{s}_{12}], \cdot)$
$\mathbf{s}_i = [p_i; q_i]$
8
3
5
6
1
12
4
2
11

Idea of Vertex Degree Kernel (VDK)



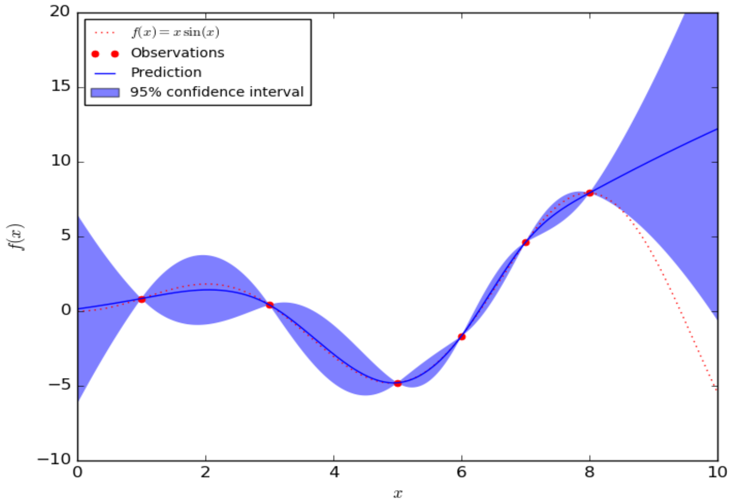
500 random trials for 500-node network
Testing 1000 out-of-sample data points.

VDK-GP with Active Learning (AL) for further gains!

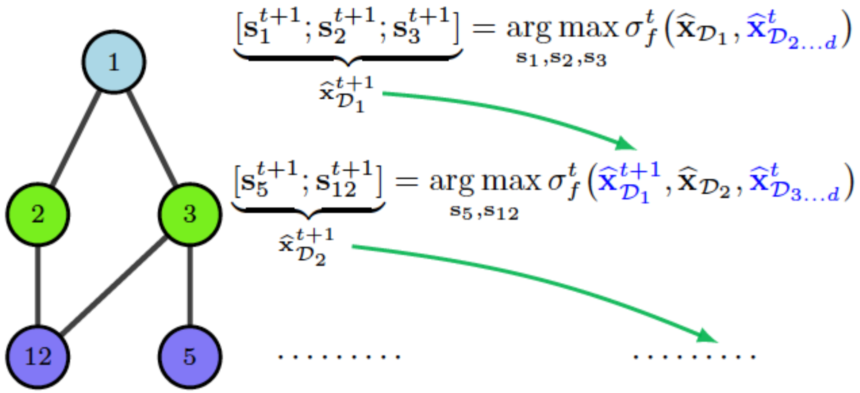
- Select training samples iteratively to maximize information gain/variance

$$\mathbf{s}^{t+1} = \arg\max_{\mathbf{s}\in\mathcal{L}} [\sigma_f^t(\mathbf{s})]^2$$

- Hard:
  - Finding maximum variance point for large-dimensional input, non-trivial



- Network–swipe Active Learning (soln)
  - Leverage VDK-GP's structure for maximizing iteratively over graph hops
  - After a few hops, no Kernel overlap between nodes (can be parallelized)
  - Low-dimensional sub-kernels: numerical optimization works fine



$$\underbrace{[\mathbf{s}_1^{t+1};\mathbf{s}_2^{t+1};\mathbf{s}_3^{t+1}]}_{\widehat{\mathbf{x}}_{\mathcal{D}_1}^{t+1}} = \arg\max_{\mathbf{s}_1,\mathbf{s}_2,\mathbf{s}_3} \sigma_f^t\left(\widehat{\mathbf{x}}_{\mathcal{D}_1}, \widehat{\mathbf{x}}_{\mathcal{D}_{2\dots d}}^t\right)$$

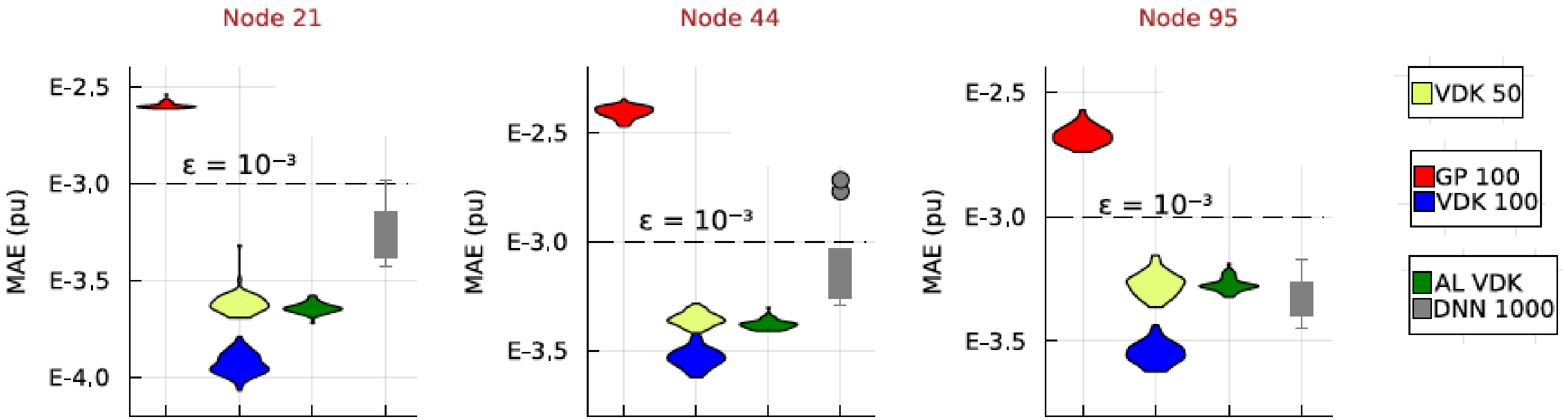
$$\underbrace{[\mathbf{s}_5^{t+1};\mathbf{s}_{12}^{t+1}]}_{\widehat{\mathbf{x}}_{\mathcal{D}_2}^{t+1}} = \arg\max_{\mathbf{s}_5,\mathbf{s}_{12}} \sigma_f^t\left(\widehat{\mathbf{x}}_{\mathcal{D}_1}^{t+1}, \widehat{\mathbf{x}}_{\mathcal{D}_2}, \widehat{\mathbf{x}}_{\mathcal{D}_{3\dots d}}^t\right)$$

$$\cdots\cdots\cdots \qquad \cdots\cdots\cdots$$

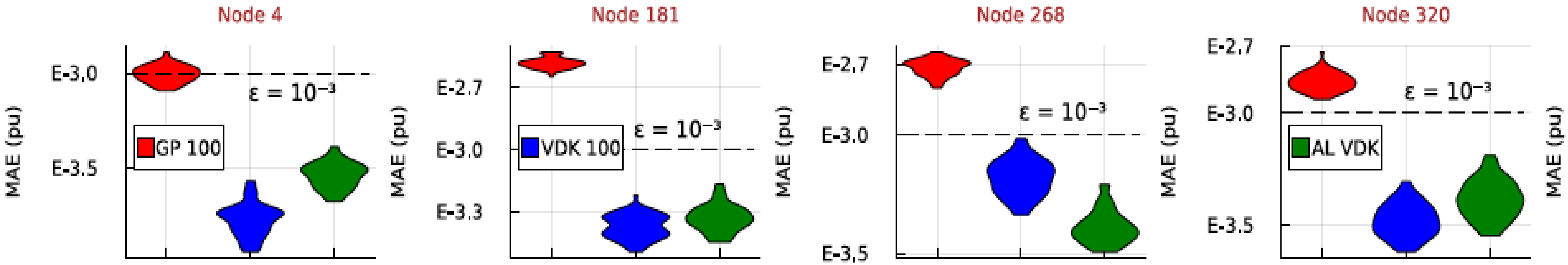
Idea of Network-Swipe Active Learning

Performance in estimating voltage in test networks

- Sample needed: GP >> VDK-GP >> AL VDK-GP (GP much better than DNN)

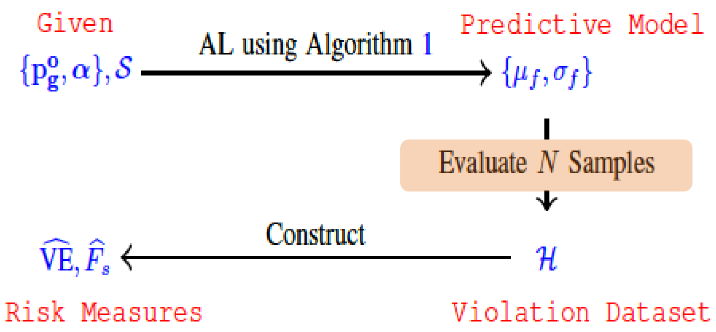


118-Bus system with 1000 out of sample data points. AL requires ~45 samples



500-Bus system with 1000 out of sample data points. AL requires ~70 samples

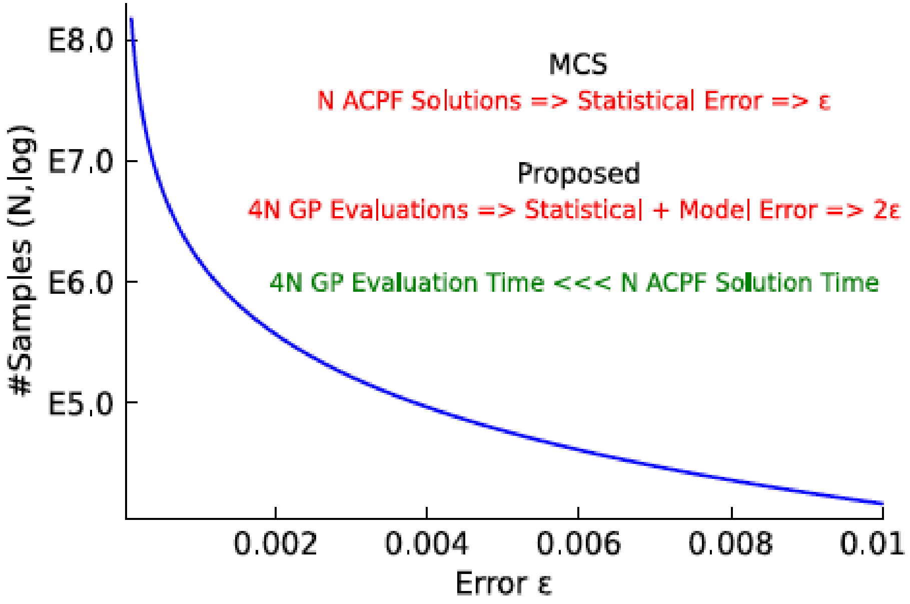
Violation Estimate (VE) using VDK-GP samples:



Theorem 1. ***Expected Value Estimation Error Bound:*** *Given that voltage values, for any two arbitrary load vectors, are jointly Gaussian. Then,* $\mathbb{P}\{|V(\mathrm{s}) - \widehat{V}(\mathrm{s})| \geq \varepsilon_m\} \leq \delta(\kappa)$ *where* $\widehat{V}(\mathrm{s}) = \mu_f(\mathrm{s}^i) \pm \kappa\sigma_f(\mathrm{s})$ *for any* $\varepsilon_m > 0$*. And with* $h(\cdot)$ *being* `Sigmoid` *function, error in VE using GP is probabilistically bounded as*

$$\mathbb{P}\Big\{\big|VE - \widehat{VE}\big| < \varepsilon_m(1 - \delta(\kappa)) + M\delta(\kappa) + \varepsilon_h\Big\} \geq 1 - \beta$$

where, $\beta \in (0,1)$, $\varepsilon_h = \sqrt{\frac{\log(2/\beta)}{2N}}$, *$M$ is a large value such that* $M > |h_p(\mathrm{s}) - h_m(\mathrm{s})|$, *and N is number of samples.*



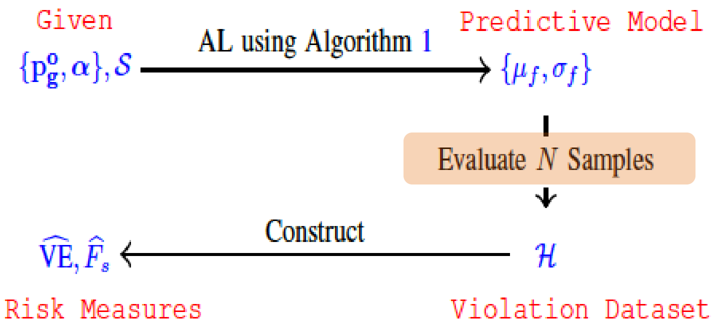
500 bus system, 95% confidence

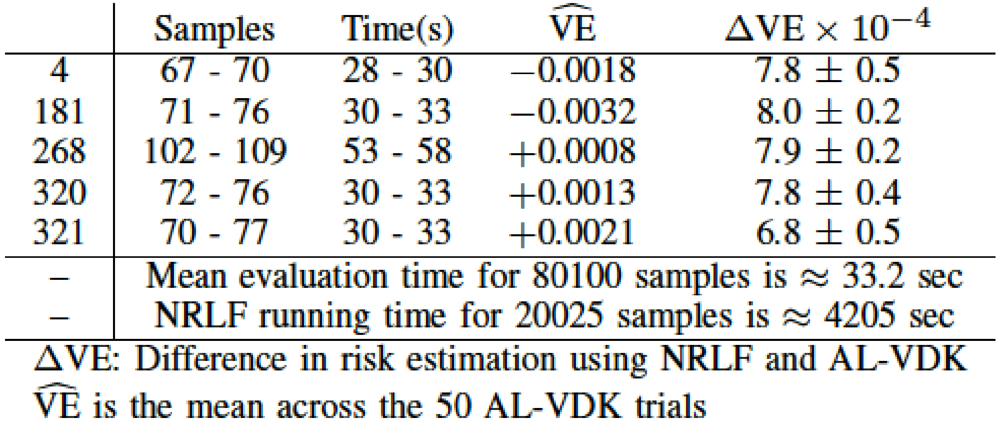
20k AC-PF solves == 4205 sec

80k GP evaluations == 33.2 sec

(120x speedup), easily within grid operator limits

Violation Estimate (VE) using VDK-GP samples:



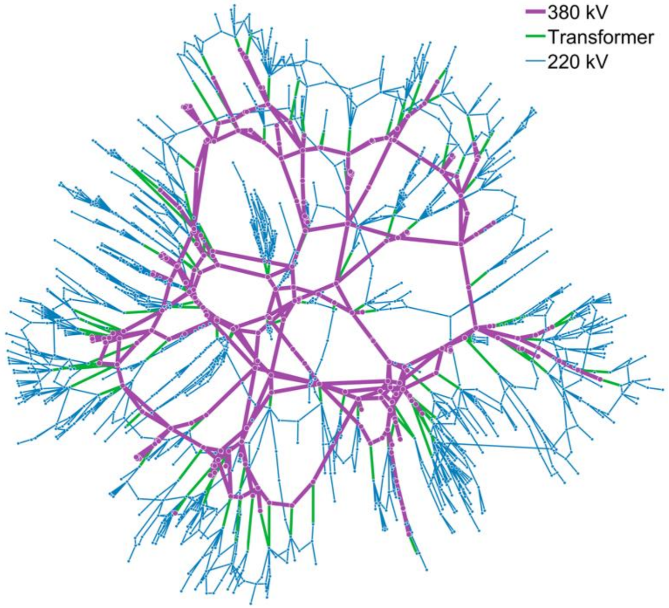


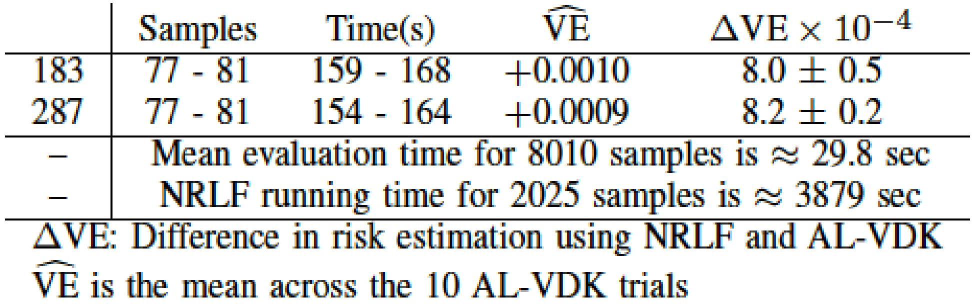
	Samples	Time(s)	$\widehat{\text{VE}}$	$\Delta\text{VE} \times 10^{-4}$
4	67 - 70	28 - 30	-0.0018	7.8 ± 0.5
181	71 - 76	30 - 33	-0.0032	8.0 ± 0.2
268	102 - 109	53 - 58	$+0.0008$	7.9 ± 0.2
320	72 - 76	30 - 33	$+0.0013$	7.8 ± 0.4
321	70 - 77	30 - 33	$+0.0021$	6.8 ± 0.5
–	Mean evaluation time for 80100 samples is ≈ 33.2 sec			
–	NRLF running time for 20025 samples is ≈ 4205 sec			

ΔVE: Difference in risk estimation using NRLF and AL-VDK

$\widehat{\text{VE}}$ is the mean across the 50 AL-VDK trials

500 node grid: 90 sec v/s 4200 sec **(45x speedup)**





	Samples	Time(s)	$\widehat{\text{VE}}$	$\Delta\text{VE} \times 10^{-4}$
183	77 - 81	159 - 168	$+0.0010$	8.0 ± 0.5
287	77 - 81	154 - 164	$+0.0009$	8.2 ± 0.2
–	Mean evaluation time for 8010 samples is ≈ 29.8 sec			
–	NRLF running time for 2025 samples is ≈ 3879 sec			

ΔVE: Difference in risk estimation using NRLF and AL-VDK

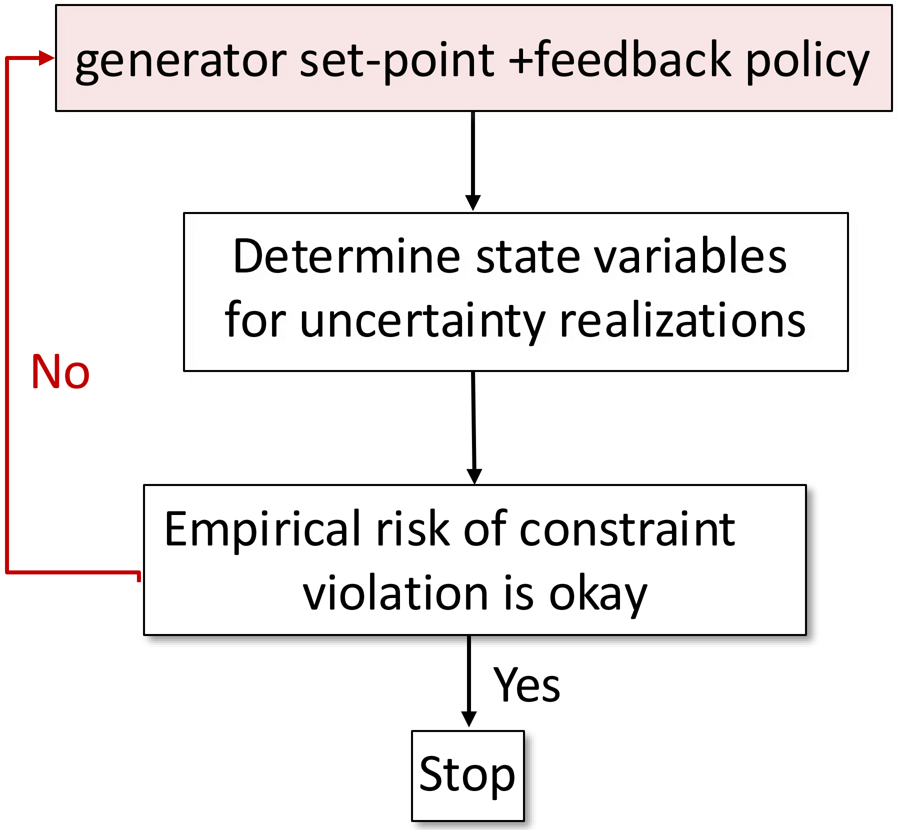
$\widehat{\text{VE}}$ is the mean across the 10 AL-VDK trials

1354 node grid: 200 sec v/s 3870 sec **(20x speedup)**

[1] P Pareek, D Deka, S Misra, Fast Risk Assessment in Power Grids through Novel Gaussian Process and Active Learning, arXiv preprint arXiv:2308.07867.

[2] P. Pareek, D. Deka, S. Misra, Data-Efficient Power Flow Learning for Network Contingencies. arXiv preprint arXiv:2310.00763.

Every 5- 15 mins:


generator set-point +feedback policy
Determine state variables for uncertainty realizations
Empirical risk of constraint violation is okay
No
Yes
Stop

Research Question:

a. Can we design ML optimization models that use limited data for solutions with confidence?

b. Can we design injection S → voltage V map for faster risk assessment using limited data?

Solution: **Bayesian** machine learning

a. Semi-supervised Bayesian Neural network for OPF

b. Network-aware Gaussian Process for voltage maps

ML proxy for OPF with *limited* training data

- Need optimal solution and constraint feasibility
- Standard ML proxies give point estimates for an input

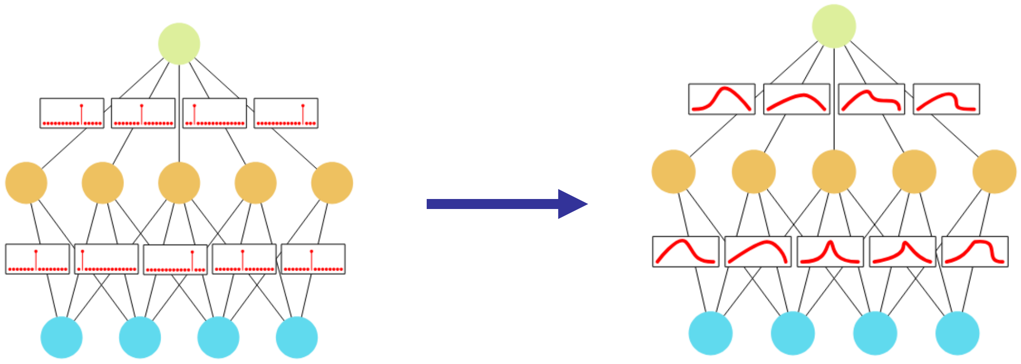
$$
\begin{aligned}
\min \quad & \sum_{g\in\mathscr{G}} c(p_g) \\
s.t. \quad & \sum_{g\in\mathscr{G}_i} p_g - P_i = \sum_{j\in\mathscr{B}} v_i v_j \left(G_{ij}\cos(\theta_i-\theta_j) + B_{ij}\sin(\theta_i-\theta_j)\right) \quad \forall i \in \mathscr{B} \\
& \sum_{g\in\mathscr{G}_i} q_g - Q_i = \sum_{j\in\mathscr{B}} v_i v_j \left(G_{ij}\sin(\theta_i-\theta_j) - B_{ij}\cos(\theta_i-\theta_j)\right) \quad \forall i \in \mathscr{B} \\
& (p, q, v, \theta) \in \mathscr{T}
\end{aligned}
$$

ML proxy for OPF with *limited* training data

- Need optimal solution and constraint feasibility
- Standard ML proxies give point estimates for an input
- Goals:
  - ✓ **Probabilistic solution with estimated confidence/variance**
  - Overcome limited labeled data (for feasibility)

$$\min_{\mathbf{y}} \; c(\mathbf{y})$$
$$\text{s.t.} \quad g(\mathbf{x},\mathbf{y}) = 0;\; h(\mathbf{x},\mathbf{y}) \leq 0$$

- Bayesian Neural Network (BNN) for OPF Proxy:



- Static weights
- Maximum Likelihood (MLE)

- Random weights (prior/posterior)
- Maximum Aposteriori (MAP)

$$p(\overbrace{\mathbf{y}|\mathbf{x}}^{\text{Labeled training data}}, w) = \textstyle\prod_i \mathcal{N}(\mathbf{y}_i | f_w(\mathbf{x}_i), \sigma_s^2)$$

$$p(w|\mathbf{x},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x},w)\, p(w)$$

Solved using variational inference (VI)

Jospin, Laurent Valentin, et al. "Hands-on Bayesian neural networks—A tutorial for deep learning users." *IEEE Computational Intelligence Magazine* 17.2 (2022): 29-48.

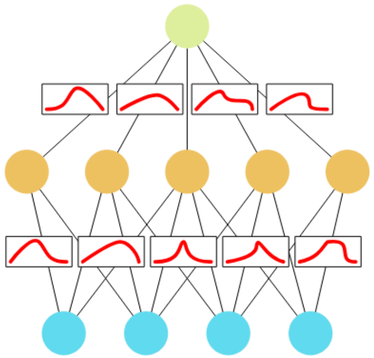
ML proxy for OPF with *limited* training data

- Need optimal solution and constraint feasibility
- Standard ML proxies give point estimates for an input
- Goals:
  - ✓ Probabilistic solution with estimated confidence/variance
  - ✓ **Overcome limited labeled data (for feasibility)**

$$\min_{\mathbf{y}} \; c(\mathbf{y})$$

$$\text{s.t.} \quad g(\mathbf{x},\mathbf{y}) = 0;\; h(\mathbf{x},\mathbf{y}) \le 0$$

- Bayesian Neural Network (BNN) for OPF Proxy:



- Random weights (prior/posterior)
- Maximum Aposteriori (MAP)

Labeled training data

$$p(w|\mathbf{x},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x},w)\,p(w)$$

$$p(\mathbf{y}|\mathbf{x},w) = \prod_i \mathcal{N}(\mathbf{y}_i|f_w(\mathbf{x}_i),\sigma_s^2)$$

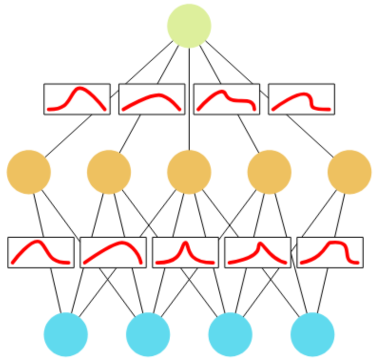
Feasibility Enhancement

(unlabeled data, true value is 0)

$$\mathcal{L}(\mathbf{y},\mathbf{x}) = \underbrace{\|g(\mathbf{x},\mathbf{y})\|^2}_{\texttt{Equality Gap}} + \underbrace{\|\text{ReLU}[h(\mathbf{x},\mathbf{y})]\|^2}_{\texttt{Inequality Gap}}$$

$$p(\mathcal{L}|\mathbf{x},w) = \prod_j \mathcal{N}\big(0|\mathcal{L}\big(f_w(\mathbf{x}_j),\mathbf{x}_j\big),\sigma_u^2\big)$$

Bayesian Neural Network (BNN) for OPF Proxy



$$\min_{\mathbf{y}} \ c(\mathbf{y}) \quad \text{s.t.} \quad g(\mathbf{x},\mathbf{y}) = 0;\ h(\mathbf{x},\mathbf{y}) \le 0$$

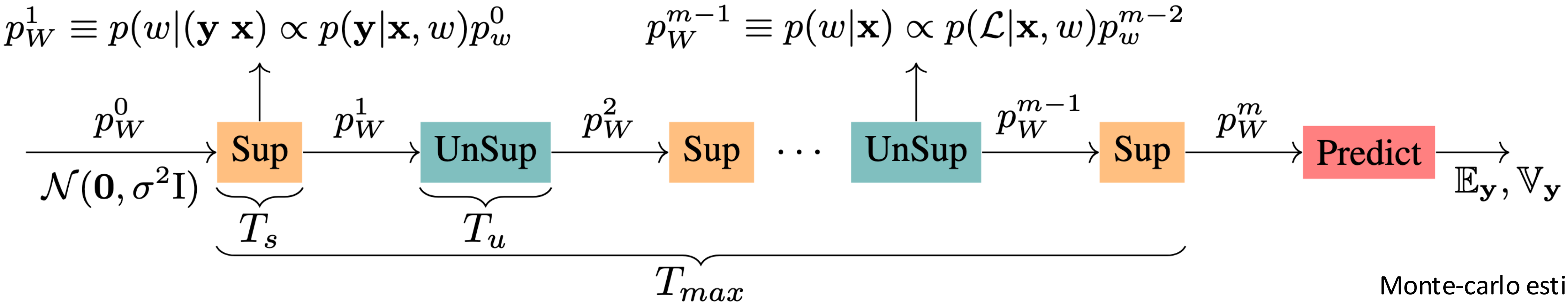
Labeled loss

Unlabeled loss

$$p(\mathbf{y}|\mathbf{x}, w) = \prod_i \mathcal{N}(\mathbf{y}_i | f_w(\mathbf{x}_i), \sigma_s^2)$$

$$p(\mathcal{L}|\mathbf{x}, w) = \prod_j \mathcal{N}(0 | \mathcal{L}(f_w(\mathbf{x}_j), \mathbf{x}_j), \sigma_u^2)$$

- Semi-supervised Sandwiched training (optimality and feasibility):



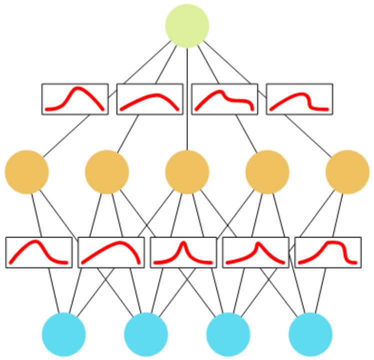
$$p_W^1 \equiv p(w|(\mathbf{y}\ \mathbf{x}) \propto p(\mathbf{y}|\mathbf{x}, w)p_w^0$$

$$p_W^{m-1} \equiv p(w|\mathbf{x}) \propto p(\mathcal{L}|\mathbf{x}, w)p_w^{m-2}$$

p_W^0
$\mathcal{N}(\mathbf{0}, \sigma^2 \mathrm{I})$
Sup
p_W^1
UnSup
p_W^2
Sup
$\cdots$
UnSup
p_W^{m-1}
Sup
p_W^m
Predict
$\mathbb{E}_{\mathbf{y}}, \mathbb{V}_{\mathbf{y}}$
T_s
T_u
T_{max}

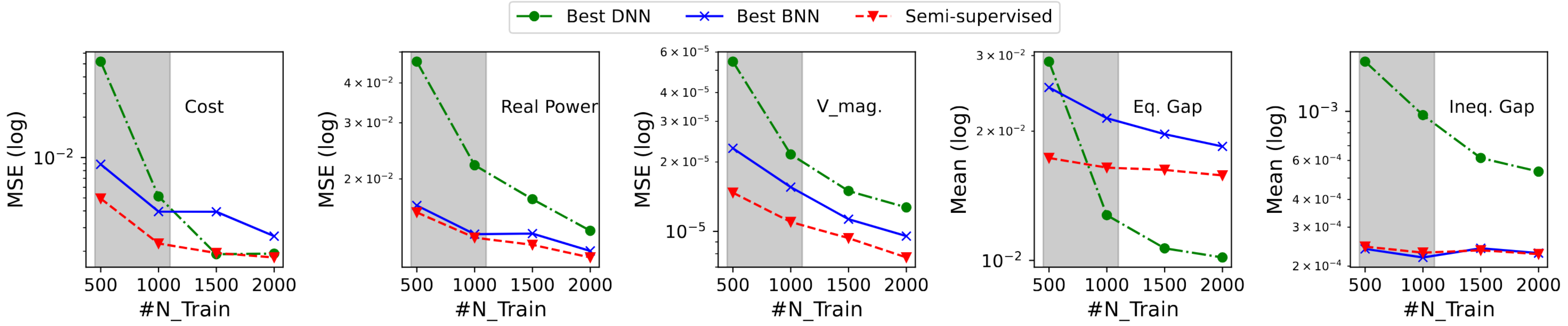
Monte-carlo estimator via sampling

Bayesian Neural Network (BNN) for OPF Proxy



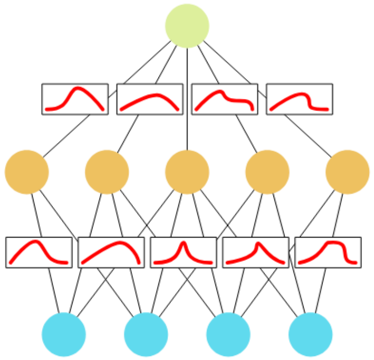
$$\min_{\mathbf{y}} \; c(\mathbf{y}) \quad \text{s.t.} \quad g(\mathbf{x},\mathbf{y}) = 0;\; h(\mathbf{x},\mathbf{y}) \leq 0$$

- Preliminary results for 57 bus system:
  - Outperforms DNN at low labelled samples and low training time



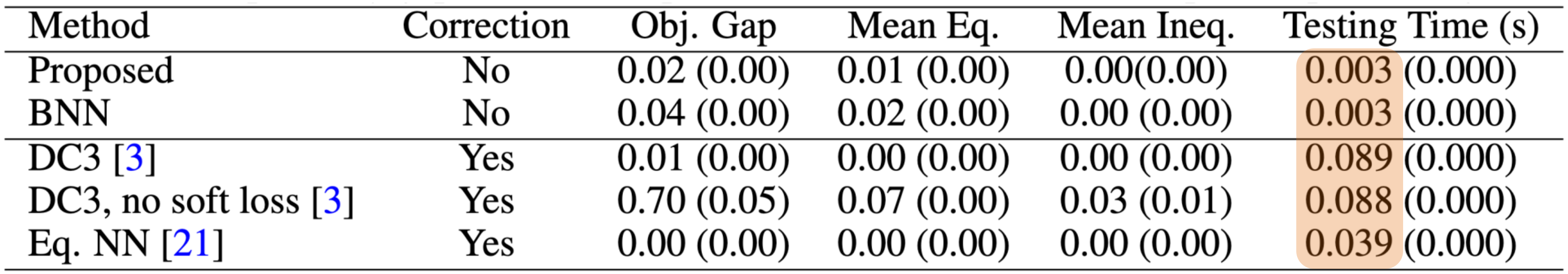
1000 sec total training, 20k unsupervised samples, BNN on Numpyro, DNN on Pytorch

Bayesian Neural Network (BNN) for OPF Proxy



$$\min_{\mathbf{y}} \; c(\mathbf{y}) \quad \text{s.t.} \quad g(\mathbf{x},\mathbf{y}) = 0;\ h(\mathbf{x},\mathbf{y}) \leq 0$$

- Preliminary results for 57 bus system:
    - Outperforms DNN at low labelled samples and low training time



Method	Correction	Obj. Gap	Mean Eq.	Mean Ineq.	Testing Time (s)
Proposed	No	0.02 (0.00)	0.01 (0.00)	0.00(0.00)	0.003 (0.000)
BNN	No	0.04 (0.00)	0.02 (0.00)	0.00 (0.00)	0.003 (0.000)
DC3 [3]	Yes	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.089 (0.000)
DC3, no soft loss [3]	Yes	0.70 (0.05)	0.07 (0.00)	0.03 (0.01)	0.088 (0.000)
Eq. NN [21]	Yes	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.039 (0.000)

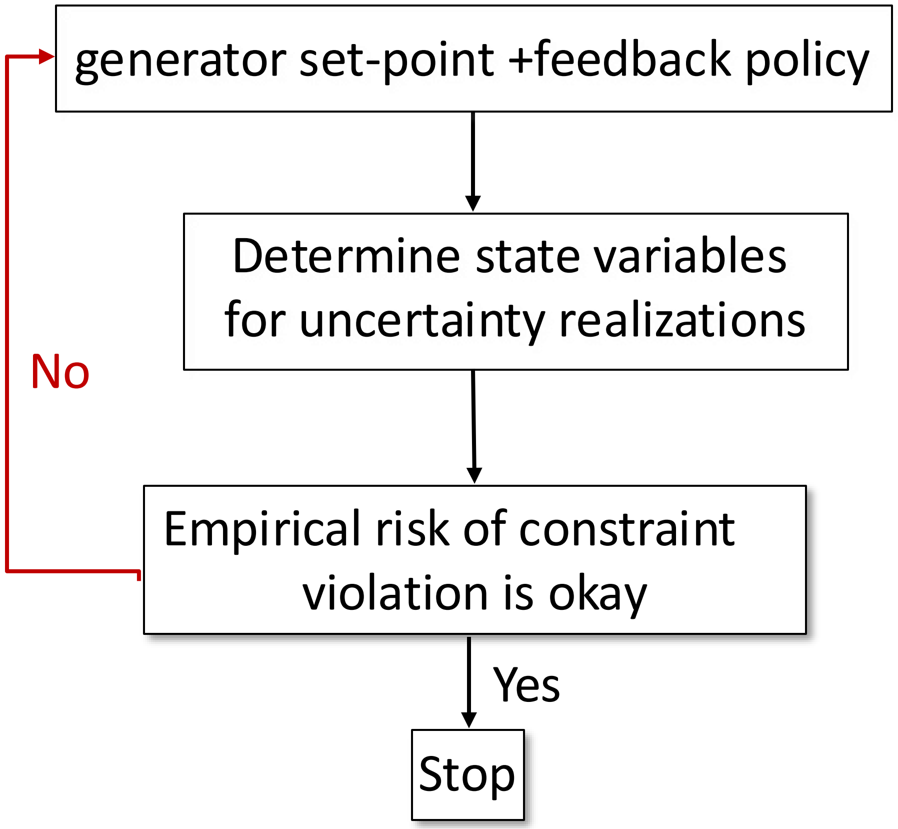
100 test instances for 57-Bus, 1000 labeled samples, 1000 sec for training, no projection in BNN

[3] P. Donti, D. Rolnick, and J. Z. Kolter. Dc3: A learning method for optimization with hard constraints. In International Conference on Learning Representations, 2021.

[21] A. S. Zamzam and K. Baker. Learning optimal solutions for extremely fast ac optimal power flow. In *2020 IEEE international conference on communications, control, and computing technologies for smart grids (SmartGridComm)*, pages 1–6. IEEE, 2020.

Next Steps:

Every 5- 15 mins:



generator set-point +feedback policy

Determine state variables for uncertainty realizations

Empirical risk of constraint violation is okay

No

Yes

Stop

a. Bayesian Neural networks for OPF

- ○ More testing
- ○ Use of confidence in follow up applications

b. Network-aware GP for voltage modeling

- ○ Use in distribution grids (limited data)
- ○ N-k applications

❖ Use BNN OPF confidence values to guide Monte Carlo or GP based validation of bounds (better than Hoeffding)

Co-authors:

Parikshit Pareek
LANL

Sidhant Misra
LANL

Kaarthik Sundar
LANL

[1] P Pareek, D Deka, S Misra, Graph-Structured Kernel Design for Power Flow Learning using Gaussian Processes, arXiv preprint arXiv:2308.07867.

[2] P. Pareek, D. Deka, S. Misra, Data-Efficient Power Flow Learning for Network Contingencies. arXiv preprint arXiv:2310.00763.

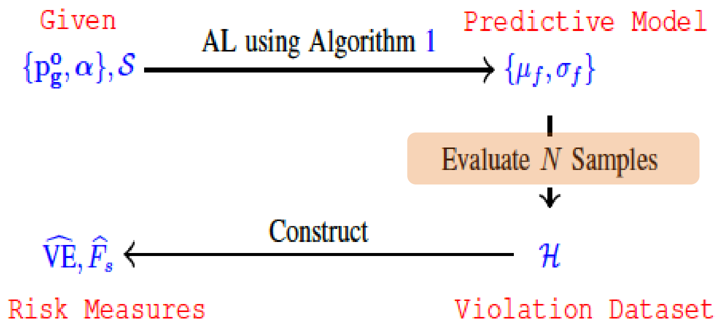
Support from:

U.S. DEPARTMENT OF ENERGY | OFFICE OF ELECTRICITY

LDRD
LABORATORY DIRECTED
RESEARCH & DEVELOPMENT

Thank You. *Questions!*

Violation Estimate (VE) using VDK-GP samples:



Given

$\{\mathrm{p}_{\mathrm{g}}^{\mathrm{o}}, \alpha\}, \mathcal{S}$

AL using Algorithm 1

Predictive Model

$\{\mu_f, \sigma_f\}$

Evaluate N Samples

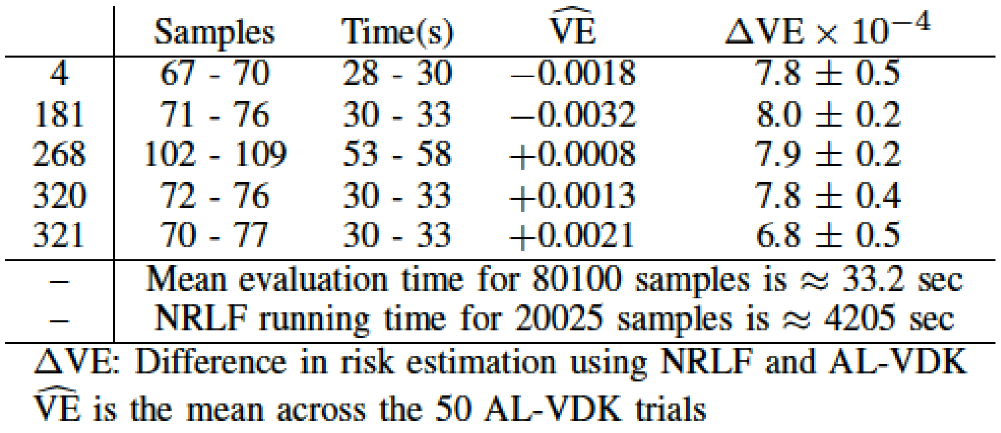
Construct

$\widehat{\mathrm{VE}}, \widehat{F}_s$

$\mathcal{H}$

Risk Measures

Violation Dataset



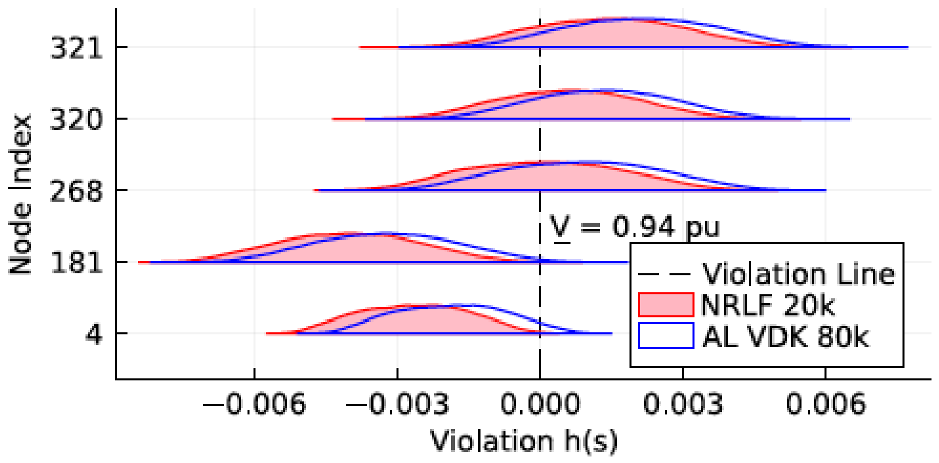
	Samples	Time(s)	$\widehat{\mathrm{VE}}$	$\Delta\mathrm{VE} \times 10^{-4}$
4	67 - 70	28 - 30	-0.0018	7.8 ± 0.5
181	71 - 76	30 - 33	-0.0032	8.0 ± 0.2
268	102 - 109	53 - 58	$+0.0008$	7.9 ± 0.2
320	72 - 76	30 - 33	$+0.0013$	7.8 ± 0.4
321	70 - 77	30 - 33	$+0.0021$	6.8 ± 0.5
–	Mean evaluation time for 80100 samples is ≈ 33.2 sec			
–	NRLF running time for 20025 samples is ≈ 4205 sec			

ΔVE: Difference in risk estimation using NRLF and AL-VDK

$\widehat{\mathrm{VE}}$ is the mean across the 50 AL-VDK trials

Theorem 2. *The GP-based predictive model overestimates probability of voltage violation i.e.* $\mathbb{P}\{h(\mathbf{s}) > 0\} \geq \widehat{\mathbb{P}}\{h_m(\mathbf{s}) > 0\}$ *for* $\mathbf{s} \in \mathcal{S}$.

500 node grid


Node Index
V = 0.94 pu
Violation Line
NRLF 20k
AL VDK 80k
Violation h(s)