Physics-Informed Graph Neural Networks for Optimal Power Flow

Giuseppe Alessio D'Inverno, MathLab, SISSA
(in collaboration with Briens,D., Varbella, A., Gjorgiev, B., *RRE, ETH, Zurich*)

PINN-PAD, Padova
13/02/2024

AC-OPF Motivation

→ AC–OPF = Alternating Current Optimal Power Flow

→ Characteristics of electricity transmission:

- 100% reliable
- Demand = Load
- Hard to store

→ Consequence:

- Grid operator has to figure out how to set up generators to meet the demand every 15 minutes

Problem setup

→ Overview of the problem:

- Non-convex
- Highly Non-linear

→ Common way to solve the problem: small-angle approximation

- Relaxation to DC model

→ Non-linear solvers exist - interior point method

- Long computation time for large grids → **Too long for daily grid operation!**

→ **Solution:** Machine-learning based approach

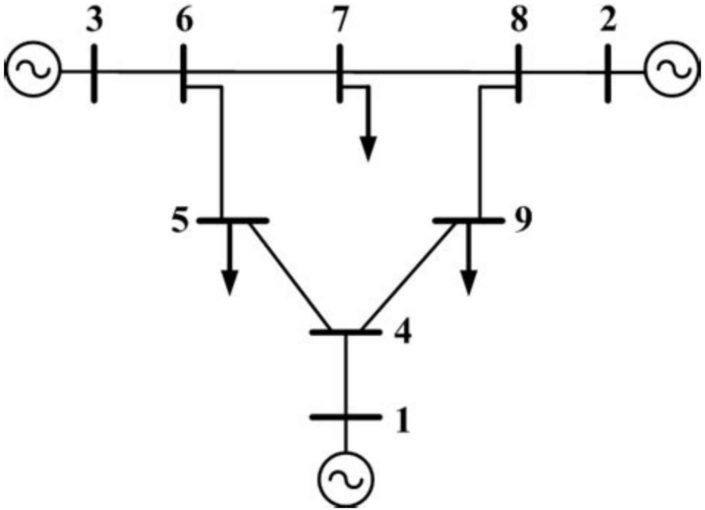
Problem setup

→ The OPF problem:

- System of generators, loads, lines and transformers
- Demand profile as an **input** - $(P_d; Q_d)$
- **How do you set up generators to meet demand with the lowest cost?**

→ AC OPF – output:

- Generated active power - P_g
- Generated reactive power - Q_g
- Voltages at each node - V
- Angle at each node - θ



IEEE – Case9 bus system

Song, Y., Hill, D. J., & Liu, T. (2015, September). Small-disturbance angle stability analysis of microgrids: A graph theory viewpoint. In *2015 IEEE Conference on Control Applications (CCA)* (pp. 201-206). IEEE.

OPF model

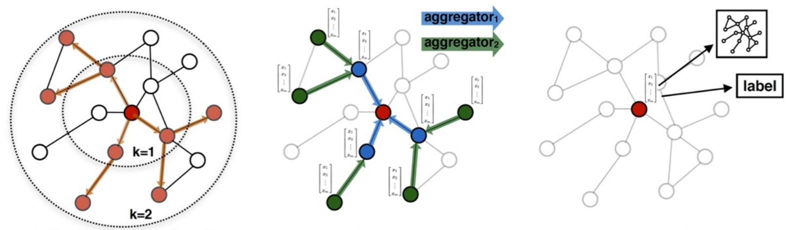
$$\text{Objective function : } \min_{P_{G,i},Q_{G,i},v_i,\theta_i,i\in V} \sum_{i\in V} C_i(P_{G,i})$$

$$\begin{cases}
\text{Power Flow equations : } \begin{cases} p^f_{ij} = g_{ij}(\tau_{ij}v_i^2) - v_i v_j(b_{ij}sin(\theta_{ij}) + g_{ij}cos(-\theta_{ij})) \\ q^f_{ij} = (-g_{ij} + \frac{Sh_{ij}}{2})(-\tau_{ij}v_i^2) - v_i v_j(g_{ij}sin(\theta_{ij}) - b_{ij}cos(\theta_{ij})) \end{cases} \\
\text{Nodal Balance : } \begin{cases} P_{G,i} - P_{D,i} - g^i_{sh} = \sum_{(ij)\in E} p^f_{ij} \\ Q_{G,i} - Q_{D,i} + b^i_{sh} = \sum_{(ij)\in E} q^f_{ij} \end{cases} \\
\text{Bus voltage magnitude limits : } v_{min,i} < v_i < v_{max,i} \forall i \in V \\
\text{Bus active power limits : } P_{G,min,i} < P_{G,i} < P_{G,max,i} \forall i \in V \\
\text{Bus reactive power limits : } Q_{G,min,i} < Q_{G,i} < Q_{G,max,i} \forall i \in V \\
\text{Transmission limits : } (p^f_{ij})^2 + (q^f_{ij})^2 \leq S_{max,ij}
\end{cases}$$

Graph Neural Networks (GNNs)

- Class of learners designed to process graphs (e.g. networks, molecules) and perform graph/node classification, regression, generation etc.
- Message Passing Graph Neural Networks (MP-GNNs): learning phase carried on by an updating scheme for each node:

$$\mathbf{h}_v^k = \text{COMBINE}\Big(\mathbf{h}_v^{k-1}, \text{AGGREGATE}(\{\{\mathbf{h}_u^{k-1}, u \in \text{ne}[v]\}\})\Big)$$

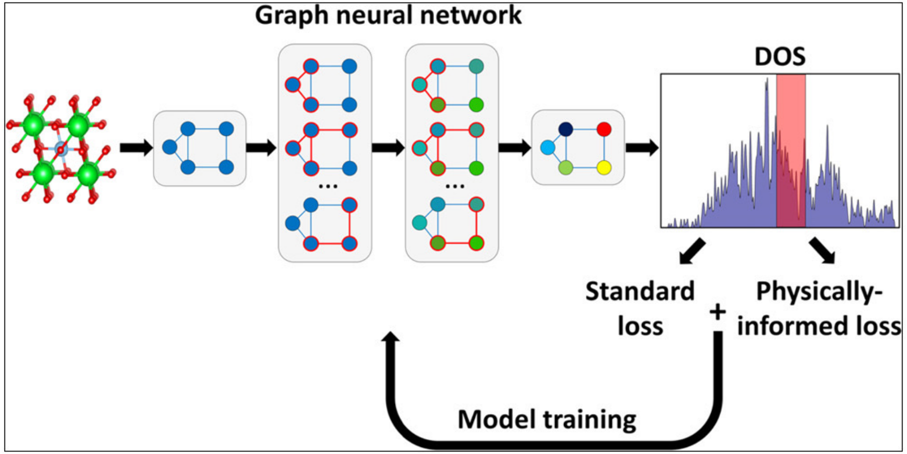


1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30*.

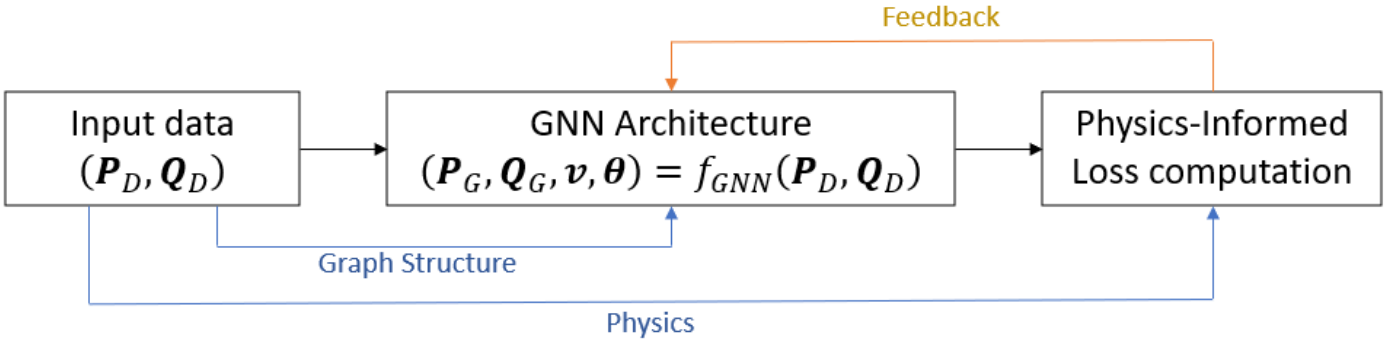
PI-GNNs

- → Combination of Physics Informed Neural Networks (PINNs) and GNNs
- → The network is "informed" of the underlying physics in the residual term of the loss function



Fung, V., Ganesh, P., & Sumpter, B. G. (2022).
Physically informed machine learning prediction of electronic density of states. *Chemistry of Materials*, *34*(11), 4848-4855.

Overview of the model



Components of the physics-informed loss

Name	**Type of Constraint**	**Mathematical Formulation**
Cost loss	Objective function	$Cost = \sum_{i \leq N} C_{P,i}(P_{G,i}) + \sum_{i \leq N} C_{Q,i}(Q_{G,i})$
Equality loss	Equality constraint	$\begin{cases} P_{G,i} - P_{D,i} - \sum_{(ij) \in E} p_{ij}^f = 0 \\ Q_{G,i} - Q_{D,i} - \sum_{(ij) \in E} q_{ij}^f = 0 \end{cases}$
Inequality loss	Inequality constraint	$\begin{cases} v_{\min,i} < v_i < v_{\max,i} \quad \forall i \in V \\ P_{G,\min,i} < P_{G,i} < P_{G,\max,i} \quad \forall i \in V \\ Q_{G,\min,i} < Q_{G,i} < Q_{G,\max,i} \end{cases}$
Flow loss	Inequality constraint	$(p_{ij}^f)^2 + (q_{ij}^f)^2 \leq S_{\max,ij}$
Plate loss	Inequality constraint	$\sum_{i \leq N} P_{G,i} - \sum_{i \leq N} P_{D,i} \geq 0$

Training: Penalthy method

- General constrained Optimization problem:

$$\min_x f(x)$$

$$subject\ to: \begin{cases} g_i(x) \le 0, i = 1, \dots, I \\ h_j(x) = 0, j = 1, \dots, J \end{cases}$$

- Conversion into a sequence of unconstrained problems with varying coefficients
  → should converge to original constrained problem

$$\min_x f(x) + \mu_g^k \sum_{i \le I} c\big(g_i(x)\big) + \mu_j^k \sum_{j \le J} h_j^2(x)$$

With: - $c\big(g_i(x)\big) = \max\left(0, g_i^2(x)\right)$

- $\begin{cases} \mu_h^{k+1} = \mu_h^k \beta_h, \ \mu_h^0 = \alpha_h \\ \mu_g^{k+1} = \mu_g^k \beta_g, \ \mu_g^0 = \alpha_g \end{cases}$ The penalty coefficients at the k-th iteration

Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., & Johnson, S. G. (2021). Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, *43*(6), B1105-B1132.

Training: Augmented Lagrangian (AL) method

- Addition of extra terms to make the problem locally convex

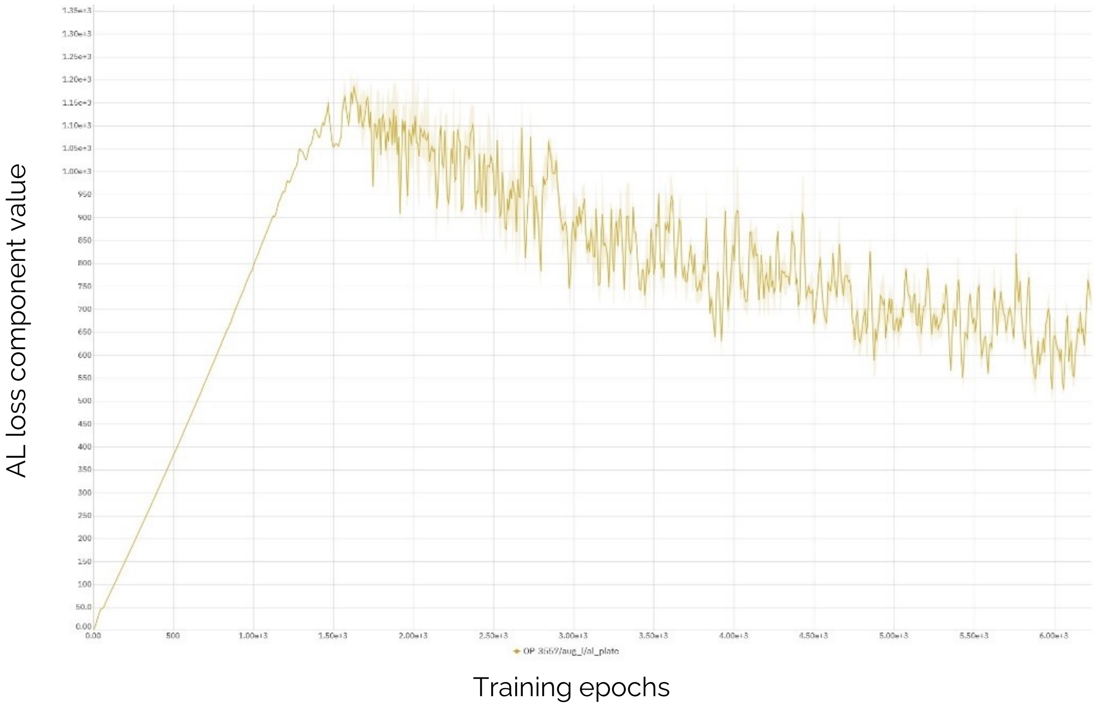
$$\min_x f(x) + \mu_g^k \sum_{l \le L} c(g_i(x)) + \mu_h^k \sum_{j \le J} h_j^2(x) + \sum_{l \le L} \lambda_{g,l}^k g_l(x) + \sum_{j \le J} \lambda_{h,j}^k h_j(x)$$

- AL coefficient associated to each constraint.
  - $\lambda_{g,l}^{k+1} = \lambda_{g,l}^k + 2\mu_g^k g_l(x^k)$
  - $\lambda_{h,j}^{k+1} = \lambda_{h,j}^k + 2\mu_h^k h_j(x^k)$

- Loss-dependent value → Stagnant loss component will increase proportionally

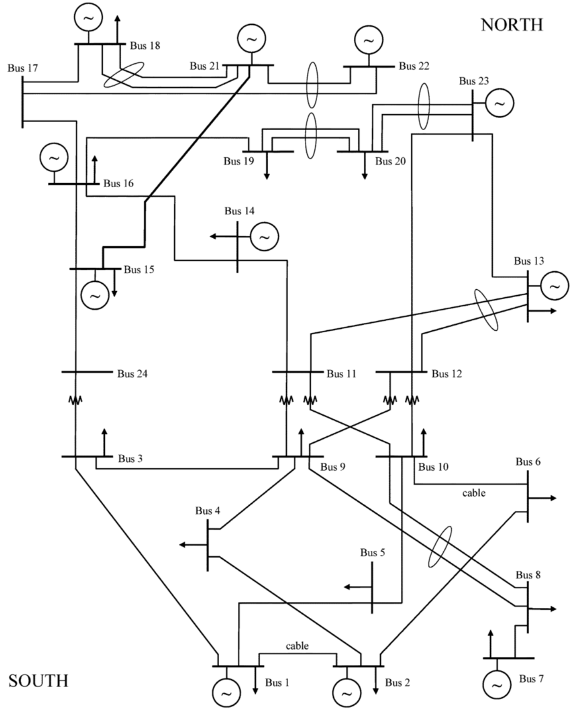
Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., & Johnson, S. G. (2021). Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, *43*(6), B1105-B1132.

Example of AL method in action



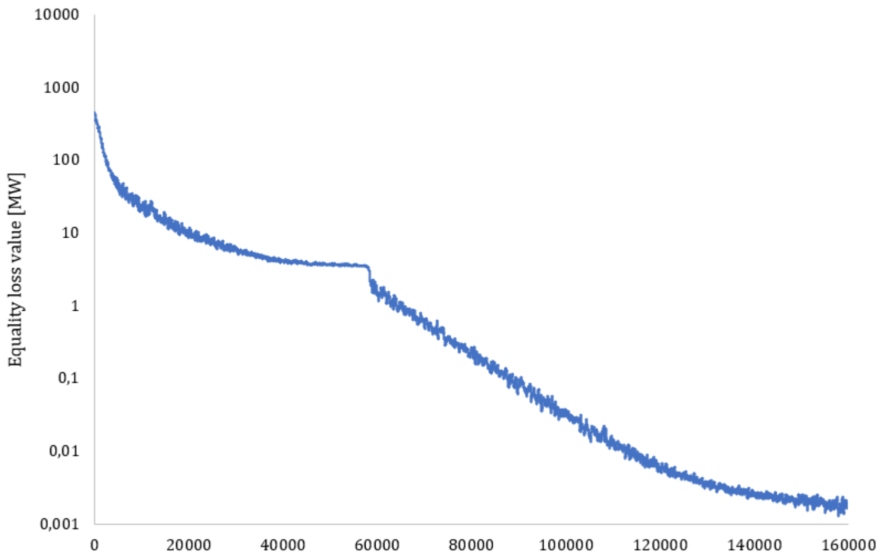
IEEE benchmark cases

- Grid systems with loads, generator, empty buses, transmission lines, transformers, shunt elements
- Benchmark cases used in literature
- Cases used:
  - Case9
  - Case9Q
  - Case24_ieee_rts
  - Case30
  - Case118



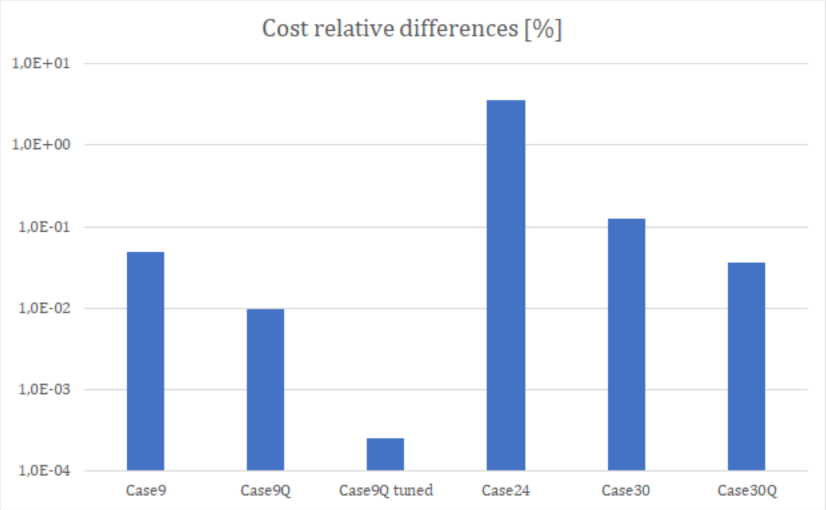
Training Regiment

- Initial testing: $M = 1$ input
  - Easy validation with solver result
  - Necessary for $M > 1$



Evolution of equality loss (nodal balance) at each epoch

Evaluation results – $M = 1$



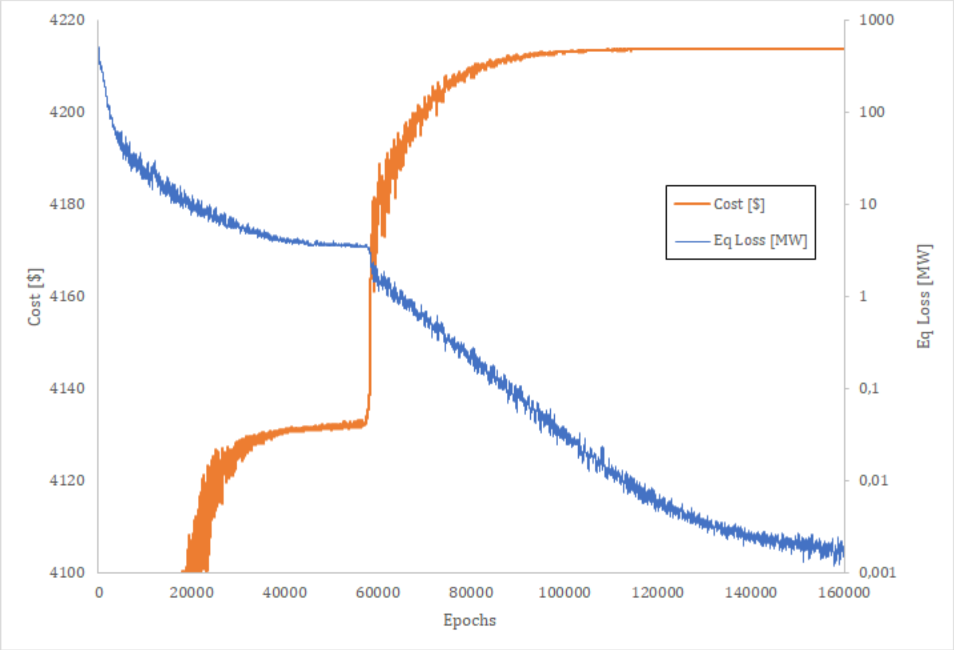
Case	Eq Loss difference [MW]
Case9	0.00010
Case9Q	0.0015
Case9Q tuned	0.00079
Case24	-6.58
Case30	-0.0054
Case30Q	0.012

Evaluation results – $M > 1$

- Training with M=20 → varying demand profile
- Test for Case9Q (tuned)

Loss Component	Absolute Loss	Relative Loss [%]
Average Equality Loss [MW]	1.50	0.48
Average Cost Difference [$]	10.52	0.44

Trade-off between cost and PF equations



Conclusions and Future works

→ PI-GNNs performances comparable to that of modern solvers

→ Great flexibility (no need to adapt hyperparameters)

Future directions:

→ Full sensitivity analysis (optimal architecture & hyperparameters)

→ Extension to connected sequences of demands (time-series)

A little advertisement…

Thanks for your attention!!

gdinvern@sissa.it
https://aledinve.github.io
giuseppe-alessio-dinverno
AleDinve
@AleDinve