Leveraging Power Grid Topology in Machine Learning Assisted Optimal Power Flow

Thomas Falconer and Letif Mones

***Abstract*—Machine learning assisted optimal power flow (OPF) aims to reduce the computational complexity of these non-linear and non-convex constrained optimization problems by consigning expensive (online) optimization to offline training. The majority of work in this area typically employs fully connected neural networks (FCNN). However, recently convolutional (CNN) and graph (GNN) neural networks have also been investigated, in effort to exploit topological information within the power grid. Although promising results have been obtained, there lacks a systematic comparison between these architectures throughout literature. Accordingly, we introduce a concise framework for generalizing methods for machine learning assisted OPF and assess the performance of a variety of FCNN, CNN and GNN models for two fundamental approaches in this domain: regression (predicting optimal generator set-points) and classification (predicting the active set of constraints). For several synthetic power grids with interconnected utilities, we show that locality properties between feature and target variables are scarce and subsequently demonstrate marginal utility of applying CNN and GNN architectures compared to FCNN for a fixed grid topology. However, with variable topology (for instance, modeling transmission line contingency), GNN models are able to straightforwardly take the change of topological information into account and outperform both FCNN and CNN models.**

***Index Terms*—OPF, graph theory, neural networks.**

Nomenclature

Functions and operators

Φ, Ψ — OPF operators that map grid parameters to optimal values of the primal variables and both primal and dual variables, respectively.

F — OPF function introduced to simplify notation of the related operator whereby only grid parameters vary.

f — Objective function of a particular OPF problem.

l — Loss function used to optimize neural network parameters, θ.

Sets

$\mathcal{A}$ — Set of active inequality constraints (those satisfied with equality at the optimal point).

$\mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}$ — Full sets of equality and inequality constraints for a particular OPF problem, respectively.

$\mathcal{F}_{\Phi}$ — Set of feasible points for the optimization variables.

$\mathcal{M}$ — Full set of neural network models for which predictive performance is assessed.

$\mathcal{N}, \mathcal{E}$ — Sets of nodes (vertices) and edges that define an undirected graph, $\mathbb{G}$, respectively.

$\mathcal{V}$ — Set of violated inequality constraints associated with a vector of optimization variables, y.

Ω — Abstract set representing the OPF operator domain.

σ — Set of hyperparameters used to define neural network architectures.

θ — Set of neural network parameters optimized during the model training process.

Variables

P_g, P_l — Power injection and withdrawal for a particular generator and load, respectively (active power components).

V_m — Bus voltage magnitude.

x — Vector of grid parameters (e.g. active and reactive power components of loads).

y — Vector of primal variables (e.g. voltage magnitudes and active power component of generator injections).

z — Vector of dual variables (Lagrangian multipliers) of the associated equality and inequality constraints.

Z_{ij} — Impedance of transmission line between bus i and bus j.

I. Introduction

OPTIMAL power flow (OPF) is an umbrella term for a family of constrained optimization problems that govern electricity market dynamics and facilitate effective planning and operation of modern power systems [1, p. 514]. Classical OPF (AC-OPF) formulates a non-linear and non-convex economic dispatch model, which minimizes the cost of generator scheduling subject to either (or both) operation and security constraints of the grid [2]. By virtue of competitive efficiency, optimal schedules are typically found using interior-point methods [3]. However, the required computation of the Hessian (second-order derivatives) of the Lagrangian at each optimization step renders a super-linear time complexity, thus large-scale systems can be prohibitively slow to solve.

This computational constraint gives rise to several challenges for independent system operators (ISOs): (1) variable inclusion

Manuscript received 3 October 2021; revised 2 March 2022 and 26 April 2022; accepted 19 June 2022. Paper no. TPWRS-01564-2021. *(Corresponding author: Thomas Falconer.)*

The authors are with the Invenia Labs, 95 Regent Street, CB2 1AW Cambridge, U.K. (e-mail: thomas.falconer@invenialabs.co.uk; letif.mones@invenialabs.co.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TPWRS.2022.3187218.

Digital Object Identifier 10.1109/TPWRS.2022.3187218

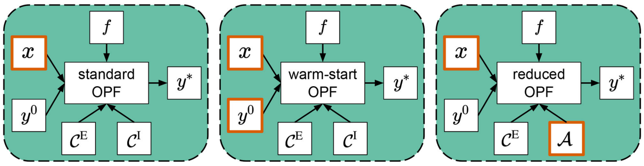


Fig. 1. Strategies for solving OPF with interior-point methods: standard (left), warm-start (center) and reduced (right) problems. x and y are the vectors of grid parameters and optimization variables, respectively, f is the objective function, $\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$ denote the sets of equality and inequality constraints, and $\mathcal{A} \subseteq \mathcal{C}^{\mathrm{I}}$ is the active subset of the inequality constraints. Typical varying arguments are highlighted in orange, whilst remaining arguments are potentially fixed.

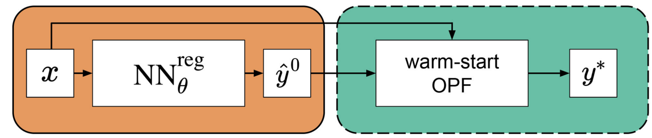


Fig. 2. Flowchart of the warm-start method (green panel) combined with a NN regressor (orange panel). For clarity, default arguments of the OPF operator are omitted.

of certain generators (i.e. unit commitment) invokes binary variables in the optimization model, thereby forming a mixed-integer, non-linear program (known to be NP-hard), exacerbating computational costs [4]; (2) the standard requirement for operators to satisfy N$-$1 security constraints (i.e. account for all contingency events where a single grid component fails) renders a much larger-scale problem, increasing the time complexity even further [5]; and lastly (3) modeling uncertainty in the supply-demand equilibrium induced by stochastic renewable generation requires methods such as scenario based Monte-Carlo simulation [6], which necessitates sequential OPF solutions at rates unattainable by conventional algorithms.

To overcome these challenges, ISOs often resort to simplified OPF models by utilizing convex relaxations [7] or linearizations [8], [9] such as the widely adopted DC-OPF model [10]. With considerably less control variables and constraints, DC-OPF can be solved very efficiently using interior-point or simplex methods [11, p. 224]. However, as DC-OPF solutions are in fact never feasible with respect to the full problem [12], set-points need to be found iteratively by manually updating the solution until convergence [13, p. 14] – hence DC-OPF is predisposed to sub-optimal generator scheduling.

In practice, ISOs typically leverage additional information about the grid in attempt to obtain solutions more efficiently. For instance, given the (reasonable) assumption that comparable grid states will correspond to neighbouring points in solution space, one can use the known solution to a similar problem as the starting value for the optimization variables of another problem – a so-called *warm-start* (Fig. 1, center panel) –, rendering considerably faster convergence compared to arbitrary initialisation [14]. Alternatively, ISOs can capitalize on the observation that only a fraction of inequality constraints are actually binding at the optimal point [15], hence one can remove a large number of constraints from the mathematical model and formulate an equivalent, but significantly cheaper, *reduced problem* [16] (Fig. 1, right panel). Security risks associated with the omission of violated constraints from the reduced problem can be mitigated by iteratively solving the reduced OPF and updating the active set until all constraints of the full problem are satisfied [17].

A. Machine Learning Assisted OPF

A compelling new area of research borne from the machine learning community attempts to alleviate reliance on subpar OPF frameworks by fitting an estimator functions on historical data. The estimators are typically neural networks (NNs) owed to their demonstrated ability to model complex non-linear relationships with negligible online computation [18]. This makes it possible to obtain predictions in real-time, thereby shifting the computational expense from online optimization to offline training – and the trained model can remain sufficient for a period of time, requiring only occasional re-training.

Most of the NN-based methods for machine learning assisted OPF can be generalized as one of two approaches: 1) *end-to-end* (or direct) models, where an estimator function is used to learn a direct mapping between the grid parameters and the optimal OPF solution; and 2) *hybrid* (or indirect) techniques – a two-step approach whereby an estimator function first maps the grid parameters to some quantities, which are subsequently used as inputs to an optimization problem to find a (possibly exact) solution. Based on the actual target type, these methods can be further categorized depending on the type of predicted quantity: regression or classification.

1) Regression: By inferring OPF solutions directly, end-to-end regression methods bypass conventional solvers altogether, offering the greatest (online) computational gains [19]. However, since OPF is a constrained optimization problem, the optimal solution is not necessarily a smooth function of the inputs: changes of the binding status of constraints can lead to abrupt changes of the optimal solution. Since the number of unique sets of binding constraints increases exponentially with system size, this approach requires training on relatively large data sets in order to obtain sufficient accuracy [20]. Moreover, there is no guarantee that the inferred solution is feasible, and violation of important constraints poses severe security risks to the grid.

Instead, one can adopt a hybrid approach whereby the inferred solution of the end-to-end method is used to initialize an interior-point solver (i.e. a warm-start), which provides an optimal solution to an optimization problem equivalent to the original one (Fig. 2). Compared to default heuristics used in the conventional optimization method, an accurate initial point could theoretically reduce the number of required iterations (and so the computational cost) to reach the optimal point [21]. However, as discussed in [22], there are several practical issues which could arise, leading to limited computational gain for this technique.

2) Classification: An alternative hybrid approach leverages the aforementioned technique of formulating a reduced problem by removing non-binding inequality constraints from the mathematical model. A NN classifier is first used to predict the active set of constraints by either 1) identifying all distinct active

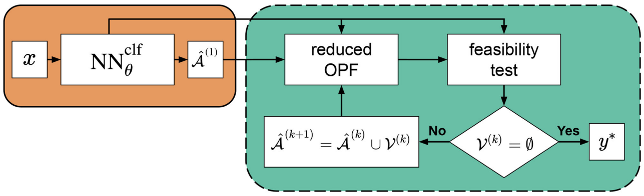


Fig. 3. Flowchart of the iterative feasibility test method (green panel) combined with a NN classifier (orange panel). $\hat{\mathcal{A}}^{(k)}$ and $\mathcal{V}^{(k)}$ are the sets of predicted active and violated inequality constraints at the k-th step of the iterative feasibility test, respectively. For clarity, default arguments of the OPF operator are omitted.

sets in the training data and using a multi-class classifier to map the features accordingly [23]; or 2) by predicting the binding status of each inequality constraint using a binary multi-label classifier [22]. Since the number of active sets increases exponentially with system size [24], the latter approach may be computationally favourable for larger grids.

To alleviate the security risks associated with imperfect classification, an *iterative feasibility test* can be employed to reinstate violated constraints until convergence, as detailed in [22] (Fig. 3). Since the reduced OPF is much cheaper relative to the full problem, this approach can in theory be rather efficient.

B. *Contributions*

Both the end-to-end and hybrid techniques for machine learning assisted OPF benefit from NN architectures designed to maximize predictive performance. Related works typically employ a range of shallow to deep fully connected neural networks (FCNN). However, convolutional (CNN) [25] and graph (GNN) [26]–[27] neural networks have recently been investigated to exploit assumed locality properties within the respective power grid, i.e. whether the topology of the electricity network influences the correlation between inputs and outputs.

Building on this set of works, our contributions are as follows:

- We introduce a concise framework for generalizing end-to-end and hybrid methods for machine learning assisted OPF by characterising them as estimators of the corresponding OPF operator or function.
- We provide a systematic comparison between the aforementioned NN architectures for both the regression and classification approaches.
- We demonstrate the marginal utility of applying CNN and GNN architectures for *fixed topology* problems (i.e. varying grid parameters only for the same topology), hence recommend the application of FCNN models for such problems.
- We show that locality properties between grid parameters (features or inputs) and corresponding generator set-points (targets or outputs) – essential for efficient inductive bias in both CNN and GNN models – are weak, which explains the moderate performance of these models compared to FCNN.
- We also show that a similar weak locality applies between grid parameters and locational marginal prices (LMPs), indicating that the applicability of CNN and GNN architectures would face similar challenges if instead used to predict these derived market signals.
- We present a set of *varying topology* problems (i.e. when both grid parameters and network topology are varied), that demonstrate successful utilization of structure based inductive bias through superior predictive performance of GNN models relative to both CNN and FCNN models.

It should be noted that, although we address the requirement of accurate predictions for machine learning assisted OPF, feasibility and optimality concerns associated with end-to-end methods, as well as the computational limitation of hybrid methods, remains a challenge for future work.

II. Methodology

A. *Problem Formulation*

This work centers on the fundamental form of OPF, without consideration for unit commitment or security constraints (although machine learning assisted OPF can be readily extended to such cases [28], [29]). In general, OPF problems can be expressed using the following concise form of mathematical programming:

$$
\begin{aligned}
\min_{y} \quad & f(x, y) \\
\text{s. t.} \quad & c_i^{\mathrm{E}}(x, y) = 0 \quad i = 1, \dots, n \\
& c_j^{\mathrm{I}}(x, y) \geq 0 \quad j = 1, \dots, m
\end{aligned}
\tag{1}
$$

where x and y are the vectors of grid parameters and optimization variables, respectively, $f(x, y)$ is the objective (or cost) function (parameterized by x), which is minimized with respect to y and subject to equality constraints $c_i^{\mathrm{E}}(x, y) \in \mathcal{C}^{\mathrm{E}}$ and inequality constraints $c_j^{\mathrm{I}}(x, y) \in \mathcal{C}^{\mathrm{I}}$. For convenience, we introduce $\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$, which denote the sets of equality and inequality constraints with corresponding cardinalities $n = |\mathcal{C}^{\mathrm{E}}|$ and $m = |\mathcal{C}^{\mathrm{I}}|$. For instance, in a simple economic dispatch problem (the focus of this work), x includes the active and reactive power components of loads, y is a vector of voltage magnitudes and active powers of generators and the objective function is a quadratic or piece-wise linear function of the (monotonically increasing) generator cost curves. Equality constraints include the power balance and power flow equations, whilst inequality constraints impose lower and upper bounds on certain quantities.

B. *OPF Operators and Functions*

By formulating the problem in such a manner as (1), one can view OPF as an operator, which maps the grid parameters (x) to the optimal value of the optimization variables (y^*) [30]. In order to introduce a consistent framework, we extend the operator arguments by the objective (f) and constraint functions ($\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$), as well as by the starting value of the optimization variables (y^0). The value of y^0 has a considerable influence of the convergence rate of interior-point methods, and for non-convex formulations with multiple possible local minima, even the found optimum is a function of y^0. The general form of the

OPF operator can be written as[1]:

$$\Phi : \Omega \rightarrow \mathbb{R}^{n_y} : \quad \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) = y^*, \tag{2}$$

where Ω is an abstract set within which the values of the operator arguments are allowed to change and n_y denotes the dimension of the optimization variables. In the simplest case, only the grid parameters vary, whilst most arguments of the OPF operator remain fixed. Accordingly, we introduce a simpler notation, the OPF function, for such cases:

$$F_\Phi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y} : \quad F_\Phi(x) = y^*, \tag{3}$$

where n_x and n_y are the dimensions of the grid parameters and optimization variables, respectively, whilst $\mathcal{F}_\Phi$ is used to denote the set of all feasible points, such that $y^* \in \mathcal{F}_\Phi$. Depending on the grid parameters, the problem may be infeasible: $\mathcal{F}_\Phi = \emptyset$.

C. Estimators of OPF Operators and Functions

Machine learning assisted OPF methods apply either an estimator operator or function, which both provide a computationally cheap prediction to the optimal point of the OPF based on the grid parameters, i.e. $\hat{\Phi}(x) = \hat{y}^* : \|\hat{y}^* - y^*\| < \varepsilon \wedge \mathbb{T}[\hat{\Phi}] \ll \mathbb{T}[\Phi]$ and $\hat{F}_\Phi(x) = \hat{y}^* : \|\hat{y}^* - y^*\| < \varepsilon \ \wedge \ \mathbb{T}[\hat{F}_\Phi] \ll \mathbb{T}[F_\Phi]$, where $\|\cdot\|$ is an arbitrary norm, ε is a threshold variable and $\mathbb{T}$ denotes the computational time to obtain the solution.

1) End-to-End: To learn the optimal OPF solution directly from the grid parameters, NNs as regressors can be used, depicted by the following function:

$$\hat{F}_\Phi(x) = \mathrm{NN}_\theta^{\mathrm{reg}}(x) = \hat{y}^*, \tag{4}$$

where subscript θ denotes the NN parameters and the superscript reg indicates that the NN is used as a regressor. The problem dimensionality can be reduced by predicting only a subset of the optimization variables – in this case, the remaining state variables can be easily obtained by solving the corresponding power flow problem [31], given the prediction is a feasible point. Optimal NN parameters can be obtained by minimizing some loss function between the ground-truth y^* and prediction $\hat{y}^*$ of some training set. Typically, the squared L2-norm, i.e. mean-squared error (MSE), is used: $\ell(y^*, \hat{y}^*) = \|y^* - \hat{y}^*\|_2^2$. To mitigate violations of certain constraints, a penalty term can be added to this loss function [20].

2) Warm-Start: Warm-start approaches utilize a hybrid model whereby a NN is first parameterized to infer an approximate set-point, $\hat{y}^0 = \mathrm{NN}_\theta^{\mathrm{reg}}(x)$, which is subsequently used to initialize the constrained optimization procedure resulting in the exact solution (y^*):

$$\begin{aligned} \hat{\Phi}^{\mathrm{warm}}(x) &= \Phi\left(x, \hat{y}^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) && (5) \\ &= \Phi\left(x, \mathrm{NN}_\theta^{\mathrm{reg}}(x), f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) && (6) \\ &= y^*. && (7) \end{aligned}$$

Optimal NN parameters can be obtained by minimizing a similar conventional loss function as in the case of the end-to-end approach. However, significant improvement has been demonstrated by optimizing NN parameters with respect to a (meta-)loss function corresponding directly to the time complexity of the entire pipeline (i.e. including the warm-started OPF) [32]: $\ell(\hat{y}^0) = \mathbb{T}[\Phi(x, \hat{y}^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}})]$.

3) Reduced Problem: In this hybrid approach, a binary multi-label NN classifier ($\mathrm{NN}_\theta^{\mathrm{clf}}$) is used to predict the active set of constraints, and a reduced OPF problem is formulated, which maintains the same objective function as the original full problem:

$$\begin{aligned} \hat{\Phi}^{\mathrm{red}}(x) &= \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \hat{\mathcal{A}}\right) && (8) \\ &= \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathrm{NN}_\theta^{\mathrm{clf}}(x)\right) && (9) \\ &= \hat{y}^*, && (10) \end{aligned}$$

where $\mathcal{A} \subseteq \mathcal{C}^{\mathrm{I}}$ is the active subset of the inequality constraints and $\hat{\mathcal{A}}$ is the predicted active set. It should also be noted that $\mathcal{C}^{\mathrm{E}} \cup \mathcal{A}$ contains all active constraints defining the specific congestion regime. In the case of a multi-label classifier, the output is a binary vector representing an enumeration of the set of non-trivial constraints, learnt by minimizing the binary cross-entropy (BCE) loss between the ground-truths represented by $\mathcal{A}$ and the predicted binding probabilities of constraints defining $\hat{\mathcal{A}}$: $\ell(\mathcal{A}, \hat{\mathcal{A}}) = -\sum_j c_j \log \hat{c}_j + (1 - c_j)\log(1 - \hat{c}_j)$. The output dimension of the multi-label classifier is reduced by removing trivial constraints (those that are always binding or non-binding in the training set) for training. We note that to formulate the subsequent reduced OPF problem, these constraints need to be reinstated before the iterative feasibility test to construct the complete active set.

Violated constraints omitted from the reduced model are retained using the aforementioned iterative feasibility test to ensure convergence to an optimal point of the full problem. The computational gain can again be further enhanced via meta-optimization by directly encoding the time complexity into a (meta-)loss function and optimizing the NN weights accordingly [22]: $\ell(\hat{\mathcal{A}}) = \mathbb{T}[\Phi(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \hat{\mathcal{A}})]$.

D. Architectures

Power grids are complex networks consisting of buses (e.g. generation points, load points etc.) connected by transmission lines, hence can conveniently be depicted as an un-directed graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denote the sets of nodes and edges (Fig. 4). Also, $\mathcal{G}$ and $\mathcal{L}$ will denote the sets of generators and loads, respectively.

This formulation motivates the use of NN architectures specifically designed to leverage the spatial dependencies within non-Euclidean data structures, i.e. GNN models – the hypothesis being that OPF problems exhibit a locality property whereby the network topology influences to correlation between grid parameters and the subsequent solution.

In real power grids, however, a given bus can include multiple generators and loads, which, although can have different power supply and demand, share the bus voltage. To accommodate such characteristics in GNN models straightforwardly, we use a

[1] We note that an even more general form of the operator can be defined when the arguments are mapped to the joint space of the primal and dual variables of the optimization problem: $\Psi : \Omega \rightarrow \mathbb{R}^{n_y + n_z} : \quad \Psi(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}) = (y^*, z^*)$, where z^* is the optimal value of the Lagrangian multipliers of the equality and inequality constraints. As locational marginal prices are computed from z^*, this formalism is useful to construct estimators for learning electricity prices.

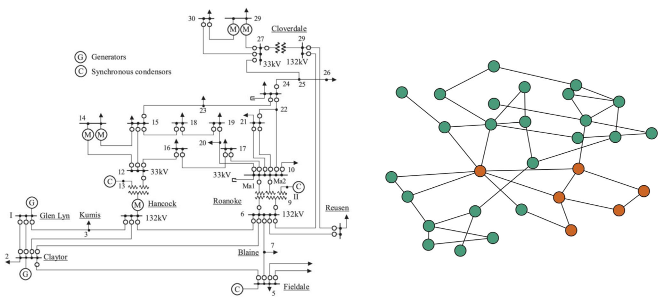


Fig. 4. Schematic diagram [33] (left) and corresponding graphical representation (right) for synthetic grid 30-ieee. Orange and green circles denote generator and load buses, respectively.

transformed version of the original graph: $\mathbb{G}' = (\mathcal{N}', \mathcal{E}')$, where each node of the transformed network represents either a single generator or a load (i.e. $|\mathcal{N}'| = |\mathcal{G}| + |\mathcal{L}|$), and generators and loads belonging to the same bus of the original network are interconnected. With this representation of the grid, generator real power outputs are obtained as individual nodal features, while bus voltage magnitudes are computed as averages of the corresponding individual voltages.

1) FCNN: Fully connected NN models, denoted by $\mathcal{M}^{\text{FCNN}}$, are used here as baseline. Their input domain is equivalent to the raw vector of grid parameters, i.e. active and reactive power components of loads: $x \in \mathbb{R}^{2|\mathcal{L}|}$, while the corresponding output vector includes the generators' injected active power and the voltage magnitude at buses comprising at least one generator ($\mathcal{N}^{\text{gen}} \in \mathcal{N}$), i.e. $y \in \mathbb{R}^{|\mathcal{G}|+|\mathcal{N}^{\text{gen}}|}$. Since FCNNs are defined in an un-structured data space, this baseline theoretically lacks sufficient relational inductive bias to efficiently exploit any underlying spatial dependencies – this information could be learnt implicitly through optimization, but possibly requires a highly flexible model with a large amount of data, thus scaling poorly to large-scale OPF problems [34]. We investigated two FCNN models using one ($\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$) and three ($\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$) hidden layers.

2) CNN: We explore the utility of augmenting the fully connected layers with an antecedent sequence of convolutional and pooling layers ($\mathcal{M}^{\text{CNN}}_{\text{global-4}}$), designed to extract a spatial hierarchy of latent features, which are subsequently (non-linearly) mapped to the target. A reasonable assumption here is that one can leverage spatial correlations within pseudo-images of the electrical grid using the weighted adjacency matrix. However, convolutions in Euclidean space are dependent upon particular geometric priors, which are not observed in the graph domain (e.g. shift-invariance), hence filters can no longer be node-agnostic and the lack of natural order means operations need to instead be permutation invariant. Nevertheless, we validate this conjecture using CNNs by combining each load constituent of length $|\mathcal{N}'|$ into a 3-dimensional tensor, i.e. $x \in \mathbb{R}^{2\times|\mathcal{N}'|\times|\mathcal{N}'|}$.

3) GNN: We analyze several GNN architectures whereby the weighted adjacency matrix is used to extract latent features by propagating information across neighbouring nodes irrespective of the input sequence [35]. Such propagation is achieved using graph convolutions, which can be broadly categorized as either spectral or spatial filtering [36].

Spectral filtering adopts methods from graph signal processing: operations occur in the Fourier domain whereby input signals are passed through parameterized functions of the normalized graph Laplacian, thereby exploiting its positive-semidefinite property. Given this procedure has $\mathcal{O}(|\mathcal{N}'|^3)$ time complexity, we investigate four spectral layers designed to reduce computational costs by avoiding full eigendecomposition of the Laplacian: (1) *ChebConv* ($\mathcal{M}^{\text{CHC}}$), which uses approximate filters derived from Chebyshev polynomials of the eigenvalues up to the K-th order [37]; (2) *GCNConv* ($\mathcal{M}^{\text{GCN}}$), which constrains the layer-wise convolution to first-order neighbours ($K = 1$), lessening overfitting to particular localities [38]; (3) *GraphConv* ($\mathcal{M}^{\text{GC}}$), which is analogous to *GCNConv* except adapting a discrete weight matrix for self-connections [39]; and (4) *GATConv* ($\mathcal{M}^{\text{GAT}}$), which extends the message passing framework of *GCNConv* by assigning each edge with relative importance through attention coefficients [40].

By contrast, spatial graph convolutions (a non-Euclidean generalization of the convolution operation found in CNNs) are performed directly in the graph domain, reducing the computational complexity whilst minimizing loss of structural information – a byproduct of reducing to embedded space [36]. We investigate *SplineConv* ($\mathcal{M}^{\text{SC}}$) [42] which, for a given node, computes a linear combination of its features together with those of its K-th order neighbours, weighted by a kernel function – the product of parameterized B-spline basis functions. The local support property of B-splines reduces the number of parameters, enhancing the computational efficiency of the operator. Note that all GNN models are named in accordance with the `PyTorch Geometric` library [43].

Finally, we note that due to the lack of connectivity information of the grid, conventional FCNN (and CNN) architectures typically fail to adapt efficiently to power system restructuring. In order to obtain sufficient performance with alternative grid topologies (i.e. contingency cases), these models need to be re-trained with appropriate training data. In contrast, GNNs compute localized convolutions in a manner such that the number of weights remains independent of the topology of the network making these models capable to train and predict on samples having different topologies [36].

E. Technical Details

1) Samples: To span multiple grid sizes, we built test cases using several synthetic grids from the Power Grid Library [44] ranging from 24 – 2853 buses. To maintain validity of the constructed data sets whilst ensuring a thorough exploration of congestion regimes, we generated 10 k (feasible) fixed topology samples for each synthetic grid by re-scaling each active and reactive load component (relative to nominal values) by factors independently drawn from a uniform distribution, $\mathcal{U}(0.8, 1.2)$. To investigate performance of the different NN architectures with varying topology, we also generated 10 k (feasible) samples subject to N$-$1 line contingency. For each sample, active and

TABLE I
NUMBER OF CHANNELS USED FOR CNN AND GNN ARCHITECTURES. σ_s AND σ_m ARE THE GRID SIZE AND MODEL TYPE BASED SCALING FACTORS. n_n DENOTES THE NUMBER OF NODES OF THE TRANSFORMED NETWORK AND n_y IS THE NUMBER OF OUTPUT VARIABLES

$$\sigma_s = \begin{cases} 1 & \text{if } |\mathcal{N}| \leq 73 \\ 2 & \text{if } |\mathcal{N}| > 73 \end{cases} \quad \sigma_m = \begin{cases} 1 & \text{if } \mathcal{M} = \mathcal{M}^{\text{GCN}} \text{ or } \mathcal{M}^{\text{GAT}} \\ 0.5 & \text{if } \mathcal{M} = \mathcal{M}^{\text{CHC}} \end{cases}$$

GNN layer	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{GNN}}_{\text{local-3}}$	$\mathcal{M}^{\text{GNN}}_{\text{global-4}}$
1.	4	$8\sigma_s$	8	8
2.	8	$16\sigma_s$	$n_n\sigma_m$	$n_n\sigma_m$
3.	16	—	$n_y\sigma_s\sigma_m$	$n_y\sigma_s\sigma_m$
Readout layer	yes	yes	no	yes

reactive load components were re-scaled as before and a single transmission line was randomly removed from the original grid topology. OPF solutions were obtained using `PowerModels.jl` [45] (an OPF package written in Julia [46]) in combination with the `IPOPT` solver [3].

2) Neural Networks: Our model with the largest number of parameters was the three hidden layer fully connected model ($\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$) that also served as the baseline. The size of each hidden layer was computed through a linear interpolation between the corresponding input and output sizes.

In the case of CNN, each model was constructed using 3×1 kernels, 1-dimensional max-pooling layers, zero-padding and a stride length of 1.

For GNN models, we investigated three architecture types: (1) the first type included two convolutional layers followed by a fully connected readout layer making the original local structure non-local ($\mathcal{M}^{\text{GNN}}_{\text{global-3}}$); (2) in the second type, only three convolutional layers were present, simply treating the features available locally at each node as the output ($\mathcal{M}^{\text{GNN}}_{\text{local-3}}$); and lastly (3) the third type was again a global one extending the above local type with a fully connected readout layer ($\mathcal{M}^{\text{GNN}}_{\text{global-4}}$). While corresponding $\mathcal{M}^{\text{GNN}}_{\text{global-3}}$ and $\mathcal{M}^{\text{GNN}}_{\text{local-3}}$ models were constructed to have an approximately equal number of parameters (details discussed below), $\mathcal{M}^{\text{GNN}}_{\text{global-4}}$ models had a significantly larger number of parameters due to the additional readout layer. For $\mathcal{M}^{\text{CHC}}$ and $\mathcal{M}^{\text{SC}}$ models, the hyperparameter K was set to 4.

Since our aim was to compare the predictive performance of models with and without topology based inductive bias, the single-layer FCNN, CNN and several GNN architectures were constructed to have a similar number of parameters for each synthetic grid. This required scaling the number of channels of the hidden layers of some architectures according to both the grid size (σ_s) and the model type (σ_m). We applied a simple grid search in order to obtain the optimal number of layers, as well as the values of parameters σ_s and σ_m. The actual number of channels used for the CNN and GNN models is presented in Table I.

Edge weights (e_{ij}) of the GNN architectures were modeled as a function of transmission line impedance, Z_{ij}, between the i-th and j-th bus. Specifically, we used the following general expression between connected buses i and j:

$$e_{ij} = \exp(-k \log |Z_{ij}|), \tag{11}$$

where k is a hyperparameter. Note that $k = 0$ leads to the application of the simple binary adjacency matrix, while in the case of $k = 1$ the absolute value of the corresponding element of the nodal admittance matrix is used.

For each grid, the generated 10 k samples were split into training, validation and test sets with a ratio of 80:10:10. In all cases, the ADAM [47] optimizer was applied (with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and learning-rate $\eta = 10^{-4}$) using an early stopping with a patience of 20 determined on the validation set. Mini-batch size of 100 was applied and hidden layers were equipped with BatchNorm [48] and a ReLU [49] activation function was used. For each model, statistics (mean and two-sided 95% confidence interval) of the predictive performance were computed using 10 independent runs.

Models were implemented in Python 3.0 using `PyTorch` [50] and `PyTorch Geometric` [43] libraries. Experiments were carried out on NVIDIA Tesla M60 GPUs. In order to facilitate research reproducibility in the field, we have made the generated samples, as well as the code our work is based upon, publicly available at `https://github.com/tdfalc/MLOPF.jl`.

III. NUMERICAL RESULTS

A. Computational Performance of Prediction

The fundamental motivation for using NN models to predict OPF solutions is their superior (online) computational performance compared to directly solving the corresponding AC-OPF problems. In Table II, we compared the average computational times of obtaining *exact* AC-OPF solutions using the `IPOPT` solver against inferring approximate solutions using various NN architectures. It is evident that, for all investigated systems, the computational time of the NN models is several orders of magnitude smaller than that of solving AC-OPF with conventional methods (note that in Table II, solve times of AC-OPF refer to a single sample, while prediction times of NN models refer to 1000 samples). Constrained optimization problems were solved on CPU (Intel Xeon E5-2686 v4, 2.3 GHz), while for the NN predictions we could utilize GPU (NVIDIA Tesla M60).

However, as discussed previously, comparing these computational times alone can be misleading: NN predictions are not necessarily optimal or even feasible. There have been several attempts to obtain feasible and possibly optimal estimates of OPF solutions (for instance by using hybrid approaches [29], [31] or introducing penalty terms of constraint violations in the loss function [20]). For all approaches, improving the quality of the predictive performance is fundamental. One apparent way is to increase the training data size significantly. In the following, we investigate the applicability of a more economical approach by using appropriate inductive bias in NN models.

TABLE II
PREDICTION TIME STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) FOR GLOBAL REGRESSION MODELS

Case	Solve time (ms)	Prediction time per 1000 samples (ms)						
	AC-OPF (IPOPT)	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	85.41 ± 1.04	10.86 ± 0.13	20.19 ± 0.11	191.75 ± 0.37	251.64 ± 2.99	196.64 ± 3.83	191.72 ± 0.75	236.52 ± 42.51
30-ieee	75.33 ± 0.63	10.58 ± 0.08	20.07 ± 0.11	194.45 ± 0.48	254.36 ± 2.16	197.59 ± 3.95	193.64 ± 0.88	237.33 ± 42.54
39-epri	147.47 ± 1.28	11.31 ± 0.14	21.47 ± 0.14	203.67 ± 1.99	269.31 ± 1.91	208.38 ± 4.43	204.75 ± 2.46	248.08 ± 43.68
57-ieee	125.24 ± 1.16	11.36 ± 0.06	21.06 ± 0.25	196.32 ± 0.27	257.18 ± 3.76	200.19 ± 4.55	196.12 ± 0.91	238.82 ± 41.02
73-ieee-rts	304.64 ± 1.32	13.25 ± 0.18	23.09 ± 0.41	216.67 ± 3.93	285.72 ± 7.28	220.83 ± 7.14	214.43 ± 3.25	260.34 ± 39.12
118-ieee	481.39 ± 2.68	12.59 ± 0.08	23.64 ± 1.94	200.02 ± 0.31	267.59 ± 3.88	203.14 ± 3.68	198.96 ± 0.29	245.38 ± 39.21
162-ieee-dtc	815.66 ± 6.27	13.81 ± 0.17	25.62 ± 2.39	207.86 ± 3.52	285.46 ± 7.57	215.16 ± 6.97	205.53 ± 3.72	261.93 ± 44.27
300-ieee	1467.43 ± 9.47	16.36 ± 0.08	28.04 ± 2.03	206.19 ± 0.74	301.14 ± 4.46	240.13 ± 3.28	203.19 ± 0.89	279.32 ± 42.01
588-sdet	2826.53 ± 51.2	22.03 ± 0.24	34.43 ± 2.36	240.94 ± 1.04	422.67 ± 3.56	363.13 ± 5.57	235.18 ± 0.64	354.07 ± 41.78
1354-pegase	10814.92 ± 29.6	36.04 ± 0.63	52.15 ± 8.89	390.56 ± 6.59	751.89 ± 9.86	676.29 ± 7.58	413.22 ± 4.71	520.68 ± 79.02
2853-sdet	34136.73 ± 99.1	76.54 ± 1.55	98.42 ± 2.42	1092.19 ± 5.79	1729.84 ± 8.66	1520.66 ± 9.32	1116.61 ± 9.94	1246.24 ± 39.39

TABLE III
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL REGRESSION MODELS (FIXED TOPOLOGY)

Case	MSE ($\times 10^{-3}$)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	0.18 ± 0.02	0.94 ± 0.04	1.55 ± 0.21	2.65 ± 0.13	$\mathbf{0.70 \pm 0.04}$	1.10 ± 0.12	1.04 ± 0.06	2.76 ± 0.19
30-ieee	0.05 ± 0.01	$\mathbf{0.03 \pm 0.01}$	0.62 ± 0.22	3.25 ± 0.82	0.09 ± 0.01	0.27 ± 0.08	0.26 ± 0.12	3.06 ± 0.33
39-epri	0.89 ± 0.10	3.16 ± 0.09	7.01 ± 0.09	4.30 ± 0.23	$\mathbf{2.38 \pm 0.10}$	3.00 ± 0.09	2.74 ± 0.13	4.72 ± 0.35
57-ieee	0.52 ± 0.11	1.62 ± 0.15	$\mathbf{1.22 \pm 0.10}$	2.18 ± 0.13	1.28 ± 0.14	1.64 ± 0.14	1.59 ± 0.14	2.28 ± 0.13
73-ieee-rts	0.21 ± 0.07	0.69 ± 0.02	1.06 ± 0.13	1.59 ± 0.11	$\mathbf{0.65 \pm 0.05}$	0.85 ± 0.11	0.85 ± 0.07	1.85 ± 0.21
118-ieee	0.39 ± 0.03	1.28 ± 0.07	3.68 ± 0.75	2.39 ± 0.12	$\mathbf{1.23 \pm 0.07}$	1.24 ± 0.07	1.27 ± 0.13	2.50 ± 0.10
162-ieee-dtc	2.61 ± 0.10	3.19 ± 0.08	3.28 ± 0.15	4.77 ± 0.21	3.08 ± 0.10	$\mathbf{2.90 \pm 0.11}$	3.04 ± 0.10	4.87 ± 0.23
300-ieee	2.06 ± 0.06	2.86 ± 0.05	3.95 ± 0.22	3.24 ± 0.09	2.42 ± 0.04	2.47 ± 0.20	$\mathbf{2.39 \pm 0.06}$	3.56 ± 0.19
588-sdet	2.56 ± 0.06	3.12 ± 0.05	4.10 ± 0.20	4.62 ± 0.36	3.25 ± 0.07	$\mathbf{3.00 \pm 0.06}$	3.05 ± 0.05	5.07 ± 0.30
1354-pegase	0.83 ± 0.12	$\mathbf{1.30 \pm 0.09}$	2.78 ± 0.23	2.16 ± 0.17	1.43 ± 0.09	1.35 ± 0.10	1.35 ± 0.12	2.51 ± 0.15
2853-sdet	5.99 ± 0.16	$\mathbf{6.87 \pm 0.05}$	15.71 ± 0.93	10.15 ± 0.58	9.70 ± 0.33	8.64 ± 0.29	8.49 ± 0.41	11.01 ± 0.46

TABLE IV
NUMBER OF PARAMETERS FOR GLOBAL REGRESSION MODELS (FIXED AND VARYING TOPOLOGY)

Case	# of parameters							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	6575	2156	1336	2303	2783	2943	2463	2353
30-ieee	4436	732	984	607	1087	1247	767	657
39-epri	7877	1580	1568	1035	1515	1675	1195	1085
57-ieee	13933	1610	1722	1047	1527	1687	1207	1097
73-ieee-rts	58677	19404	15504	18715	19195	19355	18875	18765
118-ieee	91835	25596	23160	26354	28178	28786	26962	26454
162-ieee-dtc	104396	7800	7524	8558	10382	10990	9166	8658
300-ieee	440480	82938	78006	83696	85520	86128	84304	83796
588-sdet	1512583	207152	200700	212838	214662	215270	213446	212938
1354-pegase	8486627	1408680	1390548	1409438	1411262	1411870	1410046	1409538
2853-sdet	42568525	9233926	9166558	9299404	9301228	9301836	9300012	9299504

B. *Fixed Topology*

We begin our analysis by investigating the predictive performance of NN models trained (and tested) using data derived from power grids with a fixed topology. In these experiments, only the grid parameters were varied within the datasets, while all the grid connections were the same among the samples. In this setup, FCNN and CNN architectures are functions of the grid parameters only, i.e. for regression and classification approaches we have $\text{NN}^{\text{reg}}_{\theta}(x_i) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i) = \hat{\mathcal{A}}_i$, where x_i is the grid parameter vector of the i-th sample. For GNN models, besides the grid parameters, the grid topology is also passed: $\text{NN}^{\text{reg}}_{\theta}(x_i, \mathbb{G}) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i, \mathbb{G}) = \hat{\mathcal{A}}_i$, where $\mathbb{G}$ represents the (fixed) grid topology with corresponding edge weights.

1) Regression: For each grid, Table III summarizes the MSE statistics for regression model architectures that encode the targets as global variables. The first column includes the results of our baseline $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ model, which has the largest number of parameters (Table IV). In the presence of appropriate locality attributes, CNN and GNN models are expected to provide a comparable performance to $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ with a significantly smaller amount of parameters due to their topology based inductive bias.

In order to investigate the predictive performance with and without topological information, we first constructed global

TABLE V
TRAINING TIME STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) FOR GLOBAL REGRESSION MODELS

Case	Training time ($\times 10^2$ s)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	0.75 ± 0.14	3.40 ± 0.47	$\mathbf{1.60 \pm 0.31}$	12.06 ± 1.76	20.78 ± 1.97	10.80 ± 1.87	11.81 ± 1.67	15.56 ± 2.64
30-ieee	0.57 ± 0.05	$\mathbf{0.58 \pm 0.04}$	1.07 ± 0.22	9.75 ± 3.03	16.04 ± 1.96	9.22 ± 1.30	14.06 ± 3.55	22.30 ± 6.16
39-epri	0.58 ± 0.10	$\mathbf{0.82 \pm 0.05}$	0.83 ± 0.17	12.23 ± 1.70	16.36 ± 2.90	8.69 ± 1.17	9.70 ± 0.74	15.66 ± 3.34
57-ieee	0.33 ± 0.08	$\mathbf{0.67 \pm 0.03}$	1.13 ± 0.17	12.73 ± 1.83	12.39 ± 2.99	9.20 ± 2.33	11.93 ± 2.10	13.69 ± 2.22
73-ieee-rts	0.83 ± 0.15	2.79 ± 0.12	$\mathbf{1.64 \pm 0.21}$	12.36 ± 1.82	19.13 ± 2.19	10.07 ± 1.49	12.29 ± 1.92	16.36 ± 2.53
118-ieee	0.43 ± 0.09	1.98 ± 0.18	$\mathbf{1.66 \pm 0.28}$	17.80 ± 2.44	8.25 ± 0.96	7.10 ± 0.77	5.73 ± 0.53	20.62 ± 1.99
162-ieee-dtc	0.28 ± 0.04	1.32 ± 0.17	$\mathbf{1.08 \pm 0.23}$	14.13 ± 2.56	6.45 ± 0.83	8.49 ± 1.69	7.44 ± 1.29	12.19 ± 1.89
300-ieee	0.33 ± 0.02	$\mathbf{0.64 \pm 0.05}$	1.70 ± 0.27	14.74 ± 1.94	11.87 ± 1.01	13.25 ± 1.63	8.43 ± 1.23	16.91 ± 5.28
588-sdet	0.65 ± 0.15	$\mathbf{0.58 \pm 0.05}$	1.84 ± 0.40	23.74 ± 6.36	11.24 ± 1.26	15.54 ± 3.02	10.72 ± 1.63	22.61 ± 4.16
1354-pegase	1.81 ± 0.22	$\mathbf{1.13 \pm 0.11}$	1.52 ± 0.43	18.07 ± 3.34	22.55 ± 1.46	26.74 ± 5.08	13.74 ± 1.77	21.54 ± 2.84
2853-sdet	9.54 ± 0.44	1.37 ± 0.05	$\mathbf{0.54 \pm 0.02}$	14.72 ± 1.00	16.93 ± 0.88	24.35 ± 2.19	14.38 ± 1.24	17.29 ± 3.29

TABLE VI
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (FIXED TOPOLOGY)

Case	MSE ($\times 10^{-3}$)					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	73.93 ± 8.46	$\mathbf{27.03 \pm 0.36}$	63.69 ± 9.76	2.63 ± 0.12	$\mathbf{0.50 \pm 0.04}$	2.48 ± 0.12
30-ieee	29.83 ± 5.39	$\mathbf{0.23 \pm 0.05}$	19.45 ± 6.46	2.39 ± 0.12	$\mathbf{0.06 \pm 0.01}$	2.84 ± 0.13
39-epri	14.46 ± 2.84	$\mathbf{3.27 \pm 0.18}$	15.09 ± 2.92	2.81 ± 0.14	$\mathbf{2.11 \pm 0.07}$	3.24 ± 0.19
57-ieee	8.53 ± 3.65	$\mathbf{2.29 \pm 0.15}$	9.80 ± 4.50	2.14 ± 0.15	$\mathbf{1.09 \pm 0.17}$	2.35 ± 0.22
73-ieee-rts	36.85 ± 1.53	$\mathbf{31.69 \pm 0.11}$	53.01 ± 1.03	1.31 ± 0.14	$\mathbf{0.35 \pm 0.04}$	1.67 ± 0.13
118-ieee	31.57 ± 3.29	$\mathbf{6.47 \pm 0.20}$	39.85 ± 7.85	3.91 ± 0.09	$\mathbf{1.41 \pm 0.09}$	4.34 ± 0.27
162-ieee-dtc	11.71 ± 0.61	$\mathbf{6.27 \pm 0.18}$	11.81 ± 0.60	6.40 ± 0.12	$\mathbf{3.47 \pm 0.11}$	5.55 ± 0.14
300-ieee	16.79 ± 2.59	$\mathbf{9.35 \pm 0.15}$	46.63 ± 8.50	3.48 ± 0.08	$\mathbf{2.83 \pm 0.08}$	5.01 ± 1.34
588-sdet	19.98 ± 2.27	$\mathbf{16.30 \pm 0.24}$	22.48 ± 0.95	5.64 ± 0.18	$\mathbf{4.20 \pm 0.07}$	15.51 ± 2.25

FCNN ($\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$), CNN ($\mathcal{M}^{\text{CNN}}_{\text{global-4}}$) and GNN ($\mathcal{M}^{\text{GNN}}_{\text{global-3}}$) models in a manner such that they have a similar number of parameters for each grid (Table IV).

In general, the regression performance of the investigated models (including the baseline) has a week correlation with the system size. This indicates that other factors, for instance the actual number of active sets, can also play an important role (as observed previously in [22]).

Comparing the CNN and GNN models, we found that in most of the cases, GNN models outperform the CNN model. An interesting exception is case 57-ieee, where the CNN model appeared to perform best. However, we rather consider this as an anomalous case, where the reduced error could be attributed to the coincidental unearthing of structural information within the receptive fields when convolving over the pseudo-image of the grid.

Although GCN is the simplest GNN model we investigated, in general it performs similarly to the more sophisticated GAT model. Whilst CHC and SC models have similar performance, computational efficiencies with respect to the training times of CHC (Table V) allude to a better scaling to larger grids.

The most striking observation is that the single-layer FCNN model exhibits exceedingly comparable performance to the best GNN models. For several cases, the difference between the average MSE values of the best GNN model and the single-layer model is not statistically significant and for the two largest grids, FCNN even outperforms all GNN models. It is also worth mentioning that $\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$ has at least one order of magnitude shorter training times than the global GNN models (Table V). For many cases, the significantly larger $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ model had an even shorter training time than $\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$ due to the faster convergence.

The moderate performance of the global GNN models could be a result of the readout layer, which simply induces noise by arbitrarily mixing signals of nodes further away in the system. To investigate this possibility, we performed a set of experiments up to grid size of 588, this time with local architectures for the GCN, CHC and GAT models (left three columns of Table VI). Interestingly, although the number of parameters of these local models is comparable to that of the global models (Table VII), the observed performance of each of the three GNN models is considerably worse. This suggests that the main contribution to the predictive capacity actually stems from the readout layer and also indicates a potential lack of locality properties.

To further validate the above arguments, we investigated the effect of extending the local models with a readout layer, i.e. converting the local regression models to their global counterparts. We found that using the readout layer significantly improved the predictive performance for all cases (right three columns of Table VI).

One could argue that the improvement is due to the increased number of parameters, which did indeed approximately double (Table VII). However, comparing the performance of the two sets of global models, the difference seems to be marginal,

TABLE VII
NUMBER OF PARAMETERS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (FIXED AND VARYING TOPOLOGY)

Case	# of parameters					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	2796	3165	2996	6888	7257	7088
30-ieee	796	1045	900	1528	1777	1632
39-epri	1355	1629	1493	2935	3209	3073
57-ieee	1541	1935	1703	3151	3545	3313
73-ieee-rts	20583	21451	21145	57411	58279	57973
118-ieee	27912	28835	28600	53508	54431	54196
162-ieee-dtc	9844	10969	10284	17644	18769	18084
300-ieee	87526	89662	88698	170464	172600	171636
588-sdet	220332	224469	222260	432412	436549	434340

TABLE VIII
BCE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL CLASSIFICATION MODELS (FIXED TOPOLOGY)

Case	BCE ($\times 10^{-2}$)						
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$
24-ieee-rts	1.89 ± 0.10	3.58 ± 0.12	4.66 ± 0.52	6.93 ± 0.65	$\mathbf{3.14 \pm 0.18}$	3.52 ± 0.21	3.42 ± 0.33
30-ieee	1.71 ± 0.31	5.14 ± 0.65	4.00 ± 0.55	8.76 ± 1.24	$\mathbf{3.58 \pm 0.28}$	5.33 ± 1.21	4.98 ± 0.73
39-epri	3.61 ± 0.12	7.55 ± 0.21	13.84 ± 0.22	10.48 ± 0.31	$\mathbf{7.07 \pm 0.15}$	8.07 ± 0.26	7.60 ± 0.35
57-ieee	1.67 ± 0.14	2.51 ± 0.24	2.51 ± 0.29	2.81 ± 0.17	2.34 ± 0.18	2.24 ± 0.24	$\mathbf{2.12 \pm 0.18}$
73-ieee-rts	3.06 ± 0.14	4.34 ± 0.10	4.71 ± 0.25	6.28 ± 0.24	$\mathbf{3.34 \pm 0.11}$	4.26 ± 0.59	4.08 ± 0.89
118-ieee	4.51 ± 0.25	6.19 ± 0.21	8.29 ± 0.39	7.86 ± 0.32	4.65 ± 0.19	$\mathbf{4.35 \pm 0.21}$	4.40 ± 0.20
162-ieee-dtc	5.42 ± 0.29	6.27 ± 0.15	6.31 ± 0.34	8.32 ± 0.19	6.19 ± 0.18	$\mathbf{5.99 \pm 0.17}$	6.18 ± 0.18
300-ieee	9.32 ± 0.23	$\mathbf{8.43 \pm 0.14}$	10.97 ± 0.29	10.20 ± 0.33	8.86 ± 0.19	8.70 ± 0.16	8.65 ± 0.21
588-sdet	10.92 ± 0.22	$\mathbf{8.75 \pm 0.14}$	12.13 ± 0.45	12.14 ± 0.37	11.38 ± 0.21	11.46 ± 0.18	10.92 ± 0.14
1354-pegase	11.99 ± 0.18	$\mathbf{10.56 \pm 0.10}$	21.56 ± 0.98	17.14 ± 0.44	18.80 ± 0.32	18.43 ± 0.93	17.86 ± 0.60
2853-sdet	17.30 ± 0.36	$\mathbf{11.55 \pm 0.04}$	37.88 ± 1.59	28.58 ± 0.88	31.83 ± 0.33	30.37 ± 0.53	33.47 ± 0.61

highlighting again the utility of the fully connected component and confirming our suspicion of a lack of locality within this problem.

Finally, we also investigated the utility of using the nodal admittance matrix to express electrical distances within the power grid – i.e. setting $k = 1$ in (11) –, rather than the simple binary adjacency matrix ($k = 0$). For this inherently more sophisticated approach, the results were in fact fairly consistent to those with $k = 0$ (a table summarising the MSE statistics for such models can be found in the *Supplementary Materials*). This is again in accordance with our suspicion that locality between input and output variables for this set of problems is rather limited, hence even more sophisticated measures of distance still cannot improve the performance of the GNNs.

2) Classification: In principle, the binding status of constraints could be predicted as nodal and edge features within a GNN framework. However, based on our findings for the regression experiments (i.e. that the global strategy significantly outperforms the local one), we treated constraints only as global variables. Classification performance is reported in terms of statistics of BCE of the test set, again based on 10 independent runs (Table VIII). Additional tables concerning the number of parameters as well as the training time for each model can be found in the *Supplementary Materials*.

Here, the single-layer FCNN was observed to be even more dominant relative to the regression case. Interestingly, for larger grids, it even outperforms the three-layer FCNN, which could be suffering from over-fitting as a consequence of increased flexibility. In general, we reach a similar conclusion as in the global regression setting, whereby the performance enhancements of the GNN classifiers are marginal respective to their practicality and computational limitations. CHC and SC models perform similarly, but CHC remains the cheaper option with respect to the training time. Note that GAT was excluded from these experiments since it had already shown weak performance for the regression case relative to the other GNN models.

Although for brevity we only present the test set loss, we also note that we observed a greater precision than recall in virtually every instance. This implies that the BCE objective is more sensitive to false positives. In combination with the iterative feasibility test, which is more sensitive to false negative predictions, this can result in a significant increase in the computational cost of obtaining solutions [22]. In order to fix this misalignment, one could either use a weighted BCE (with appropriate weights for the corresponding terms) or a meta-loss objective function [22] [32].

C. Varying Topology

We now focus our analysis toward the predictive performance of NN models trained (and tested) using data derived from power

TABLE IX
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL REGRESSION MODELS WITH VARYING TOPOLOGY

Case	MSE ($\times 10^{-3}$)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	1.27 ± 0.18	1.62 ± 0.16	1.42 ± 0.17	1.91 ± 0.17	$\mathbf{0.99 \pm 0.08}$	1.42 ± 0.18	1.25 ± 0.12	1.65 ± 0.13
30-ieee	8.77 ± 0.22	8.39 ± 0.19	8.53 ± 0.18	8.68 ± 0.16	$\mathbf{0.23 \pm 0.04}$	1.92 ± 0.62	0.66 ± 0.08	3.43 ± 0.37
39-epri	12.72 ± 0.28	12.09 ± 0.22	13.33 ± 0.21	12.56 ± 0.24	$\mathbf{3.31 \pm 0.16}$	5.65 ± 0.85	4.23 ± 0.23	7.86 ± 0.33
57-ieee	4.34 ± 0.12	3.88 ± 0.13	4.01 ± 0.12	3.96 ± 0.13	$\mathbf{0.82 \pm 0.08}$	2.81 ± 0.72	1.27 ± 0.13	2.43 ± 0.16
73-ieee-rts	0.85 ± 0.05	0.95 ± 0.05	1.01 ± 0.04	1.16 ± 0.06	$\mathbf{0.66 \pm 0.07}$	0.92 ± 0.06	0.86 ± 0.07	1.24 ± 0.18
118-ieee	3.06 ± 0.14	2.59 ± 0.12	2.88 ± 0.11	2.86 ± 0.12	$\mathbf{1.15 \pm 0.05}$	1.78 ± 0.12	1.38 ± 0.08	2.66 ± 0.34
162-ieee-dtc	5.37 ± 0.18	4.38 ± 0.17	4.59 ± 0.13	5.81 ± 0.15	4.27 ± 0.13	5.29 ± 0.14	$\mathbf{3.95 \pm 0.20}$	5.29 ± 0.16
300-ieee	3.24 ± 0.08	3.16 ± 0.08	4.02 ± 0.27	3.62 ± 0.09	$\mathbf{2.42 \pm 0.07}$	2.79 ± 0.07	2.64 ± 0.06	3.72 ± 0.08
588-sdet	4.95 ± 0.12	4.02 ± 0.12	4.98 ± 0.14	4.63 ± 0.14	3.83 ± 0.31	4.16 ± 0.13	$\mathbf{3.36 \pm 0.56}$	4.46 ± 0.13

TABLE X
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (VARYING TOPOLOGY)

Case	MSE ($\times 10^{-3}$)					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	76.18 ± 8.12	$\mathbf{26.59 \pm 0.42}$	71.32 ± 9.76	1.88 ± 0.16	$\mathbf{0.74 \pm 0.12}$	1.57 ± 0.11
30-ieee	13.35 ± 2.06	$\mathbf{3.41 \pm 0.08}$	16.55 ± 6.37	8.68 ± 0.18	$\mathbf{0.15 \pm 0.02}$	2.13 ± 0.37
39-epri	46.47 ± 8.69	$\mathbf{4.47 \pm 0.22}$	24.97 ± 7.03	12.51 ± 0.21	$\mathbf{2.97 \pm 0.13}$	6.75 ± 1.06
57-ieee	9.13 ± 2.97	$\mathbf{2.09 \pm 0.23}$	24.13 ± 9.47	4.11 ± 0.12	$\mathbf{0.66 \pm 0.08}$	1.74 ± 0.21
73-ieee-rts	70.21 ± 1.84	$\mathbf{65.43 \pm 0.06}$	99.92 ± 7.04	1.23 ± 0.17	$\mathbf{0.42 \pm 0.04}$	1.23 ± 0.26
118-ieee	22.55 ± 6.68	$\mathbf{4.88 \pm 0.05}$	35.95 ± 4.88	3.02 ± 0.12	$\mathbf{1.55 \pm 0.12}$	2.97 ± 0.34
162-ieee-dtc	24.74 ± 7.34	$\mathbf{6.48 \pm 0.34}$	19.65 ± 7.61	6.38 ± 0.12	$\mathbf{4.77 \pm 0.16}$	6.57 ± 1.08
300-ieee	16.86 ± 2.55	$\mathbf{6.84 \pm 0.19}$	50.52 ± 7.97	3.68 ± 0.09	$\mathbf{3.02 \pm 0.13}$	4.67 ± 0.92
588-sdet	11.87 ± 5.38	$\mathbf{7.18 \pm 0.18}$	22.62 ± 0.19	4.89 ± 0.14	$\mathbf{4.36 \pm 0.13}$	6.96 ± 0.98

grids of size 24 – 588 with varying topology. In these experiments, we modeled the N$-$1 line contingency and samples for a given grid differed not only in their input grid parameters but also in their topology. For FCNN and CNN models, we used only grid parameters as inputs to predict the corresponding quantities of regression and classification, similarly to the fixed topology. We note that in theory, the input vector could be extended to include topological information, but it is rather cumbersome due to the quadratic scaling of the weighted adjacency matrix with system size. For GNN models, however, the change in the topology can be naturally taken into account by passing the graph information of the sample along with the grid parameters. For the regression and classification approaches we have: $\text{NN}^{\text{reg}}_{\theta}(x_i, \mathbb{G}_i) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i, \mathbb{G}_i) = \hat{\mathcal{A}}_i$, where x_i and $\mathbb{G}_i$ are the grid parameter vector and topology of the i-th sample, respectively.

1) Regression: We begin our discussion again by evaluating the global regression models (Table IX). As expected, due to the larger effective parameter space, the regression performance using samples with varying topology decreases when compared to those with fixed topology for all cases and architectures (c.f. Table III). A significant difference is that the best GNN models – CHC in most cases – outperforms both the single-layer and even the three-layer FCNN models (and CNN models too). This is resultant of the fact that in these models, any change in the network topology is ignored, whilst in the GNN architectures it is considered explicitly. This is a promising finding for applications of GNN models for predicting solutions of more sophisticated OPF problems including contingencies.

Interestingly, further investigations revealed that locality properties still play a marginal role in the predictive performance of GNNs: as for the fixed topology cases, local GNN models have a significantly weaker performance, which is subsequently restored by attaching a readout layer (Table X).

2) Classification: For the classification models, we considered again only the global case (Table XI). We note that due to the higher number of non-trivial constraints, the size of the NN models with varying topology differs from those with fixed topology (details are shown in the *Supplementary Materials*). Therefore, unlike in the case of regression, we cannot compare directly the BCE statistics of experiments with fixed and varying topology. Nevertheless, in general, we found a similar trend to the global regression, i.e. the best performing GNN model (again, most often CHC) consistently outperforms the single-layer FCNN, the CNN and even the three-layer FCNN models. This means that applying GNN models is preferable over a significantly larger FCNN architecture for both OPF related regression and classification based problems with varying topology.

D. Locality Properties

Experimental results for the NN models indicated that the general assumption of locality may not be appropriate for this problem, i.e. there is only a weak – or no existence of – locality between load inputs and generator set-point outputs. To explore this relationship further, we carried out a sensitivity analysis that directly measures locality: for each synthetic grid, we iteratively

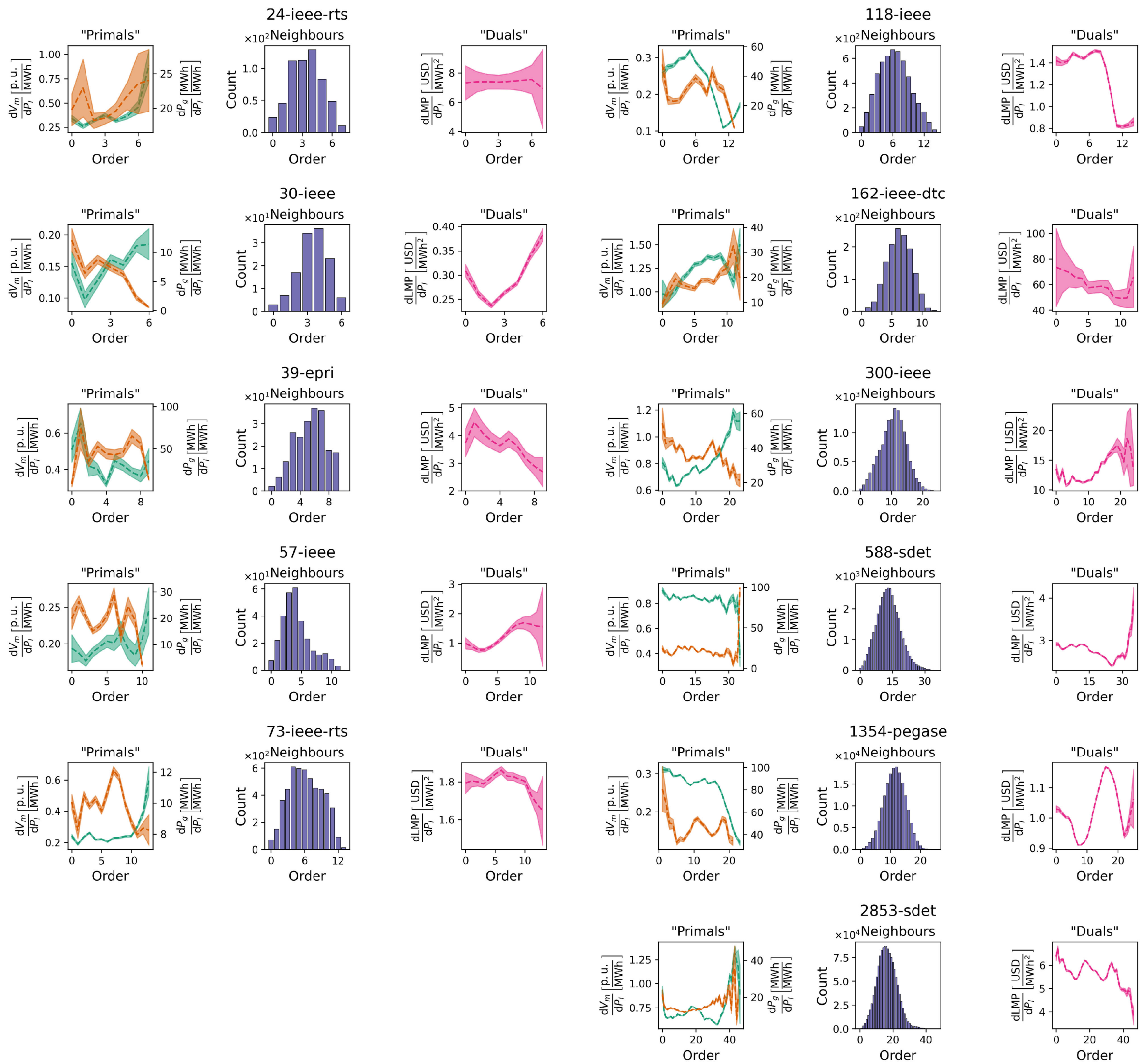


Fig. 5. Analysis of locality properties for each synthetic grid. Left and right panels show the average absolute value of the relative change (with two-sided 95% confidence intervals) in voltage magnitude (green), injected active power (orange) and locational marginal prices (purple), respectively, as a function of the topological distance from the perturbed load. Center panels show the histogram of generators with respect to the neighbourhood order from loads.

TABLE XI
BCE Statistics (Mean and Two-Sided 95% Confidence Intervals) of the Test Sets for Global Classification Models With Varying Topology

Case	BCE $(\times 10^{-2})$						
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$
24-ieee-rts	3.56 ± 0.36	3.19 ± 0.17	3.32 ± 0.18	3.43 ± 0.16	$\mathbf{1.44 \pm 0.11}$	1.81 ± 0.16	1.67 ± 0.16
30-ieee	6.21 ± 0.35	6.19 ± 0.32	6.22 ± 0.33	6.59 ± 0.37	$\mathbf{3.03 \pm 0.18}$	4.81 ± 1.35	4.43 ± 0.17
39-epri	8.66 ± 0.19	8.51 ± 0.17	9.06 ± 0.21	8.78 ± 0.21	$\mathbf{3.74 \pm 0.12}$	5.71 ± 0.86	4.36 ± 0.19
57-ieee	5.34 ± 0.24	4.56 ± 0.17	4.65 ± 0.18	4.59 ± 0.15	$\mathbf{1.88 \pm 0.07}$	3.48 ± 0.93	2.17 ± 0.09
73-ieee-rts	3.98 ± 0.22	3.87 ± 0.16	3.69 ± 0.15	4.18 ± 0.28	$\mathbf{2.25 \pm 0.12}$	2.84 ± 0.22	2.92 ± 0.21
118-ieee	4.75 ± 0.15	3.95 ± 0.11	4.27 ± 0.14	4.28 ± 0.12	2.82 ± 0.06	3.42 ± 0.14	$\mathbf{2.79 \pm 0.12}$
162-ieee-dtc	3.19 ± 0.14	2.66 ± 0.06	2.71 ± 0.06	3.23 ± 0.06	$\mathbf{2.17 \pm 0.07}$	2.65 ± 0.15	2.66 ± 0.07
300-ieee	8.07 ± 0.17	7.35 ± 0.11	7.88 ± 0.14	7.43 ± 0.17	$\mathbf{6.38 \pm 0.14}$	6.79 ± 0.14	6.74 ± 0.17
588-sdet	6.84 ± 0.77	5.91 ± 0.07	6.16 ± 0.13	5.87 ± 0.12	$\mathbf{5.12 \pm 0.08}$	6.15 ± 0.11	5.91 ± 0.08

perturbed each active load of 100 configurations by 1% and recorded the absolute value of the relative change in voltage magnitude and active power injection of each generator (i.e. $|\frac{dV_m^j}{dP_l^i}|$ and $|\frac{dP_g^j}{dP_l^i}|$, where P_l^i are the active loads with $i = 1, \ldots, |\mathcal{L}|$; and V_m^j and P_g^j are the voltage magnitude and injected active power of generators with $j = 1, \ldots, |\mathcal{G}|$), as a function of neighbourhood order (i.e. the topological distance from the perturbed load). If a grid were to exhibit locality properties, one would expect a distinct negative correlation between the average of these quantities and the respective distance from the perturbed load within the graph domain.

The results of the sensitivity analysis are shown in the left panels of Fig. 5. Although there are certain cases where either the voltage magnitude or active power injection show a weak anti-correlation with the topological distance, in general we found little evidence that the topology of the network influences the correlation between input and output variables. Plotting the distribution of generators as a function of distance from the perturbed load (middle panels of Fig. 5) suggests that this result should be of no surprise: as the system size increases, so does the average distance between the perturbed load and the generators in the system, which decreases the likelihood that nearby generators will balance corresponding demand (for apparent physical reasons such as generator capacity, line congestion etc.).

Finally, we also explored the existence of possible locality between grid inputs and the LMPs, which are functions of the duals (shadow prices) [51]. If a stronger locality property were to exist here this would be promising for using GNN models to predict electricity prices even with fixed topology [52]. However, as shown in the right panels of Fig. 5, we found no evidence of locality for the LMP values either.

IV. Conclusion

With the potential to shift the entire computational effort to offline training, machine learning assisted OPF has become an increasingly interesting research direction. Neural network based approaches are particularly promising as they can effectively model complex non-linear relationships between grid parameters and primal or dual variables of the underlying OPF problem.

Although most related works have applied fully connected neural networks so far, these networks scale relatively poorly with system size. Therefore, incorporating topological information of the electricity grid into the inductive bias of some graph neural network is a sensible step towards reducing the number of NN parameters.

In this paper, we first provided a general framework of the most widely used end-to-end and hybrid techniques and showed that they can be considered as estimators of the OPF operator or function. In this sense, our framework could be readily extended to more sophisticated OPF problems, such as consideration of unit commitment or security constraints, as well as direct prediction of derived market signals (e.g. LMPs).

We then presented a systematic comparison of several NN architectures including FCNN, CNN and GNN models. We found that for systems with fixed topology, an FCNN model has a comparable or even better predictive performance than global CNN and GNN models with similar number of parameters. The moderate performance of the CNN model can be explained by the fact that it carries out convolutions in Euclidean space (instead of the graph domain). We also identified that in the case of global GNN models, the readout layer plays a key role: constructing local models by removing their readout layer led to a significant decline in the predictive performance.

The results with fixed topology indicated that the required assumption of locality between grid parameters (inputs) and generator set-points (outputs) might not hold. To validate the findings of the NN experiments, by carrying out a sensitivity analysis we showed that locality properties are indeed scarce between grid parameters and primal variables of the OPF. Additionally, we found a similar lack of locality between grid parameters and LMPs.

Finally, we also performed a systematic comparison of NN models using varying topology of the samples. In these experiments, we modeled the N$-$1 contingency of transmission lines in both the training and test sets. We found that for such cases, global GNN architectures outperform FCNN and CNN models for both regression and classification based problems. The reason is that although locality properties still play a limited role, GNN models could take the changes of the topology into account, which were completely neglected amongst FCNN and CNN models in our setup. Although it might be possible to extend FCNN and CNN models' input by topology related features, it is definitely less straightforward than for GNN models, where this information is accounted for naturally. This property of the GNN architectures therefore makes these models promising for realistic applications, especially for security constrained OPF problems.

References

[1] A. J. Wood, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2014.

[2] R. Billinton and W. Li, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Cham, Switzerland: Springer, 1994.

[3] A. Wachter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.* vol. 106, no. 1, pp. 25–57, 2006.

[4] A. Castillo, C. Laird, C. A. Silva-Monroy, J. Watson, and R. P. O'Neill, "The unit commitment problem with ac optimal power flow constraints," *IEEE Trans. Power Syst.* vol. 31, no. 6, pp. 4853–4866, Nov. 2016.

[5] J. Rahman, C. Feng, and J. Zhang, "Machine learning-aided security constrained optimal power flow," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2020, pp. 1–5.

[6] I. Mezghani, S. Misra, and D. Deka, "Stochastic Ac Optimal Power Flow: A Data-Driven Approach," *Elect. Power Syst. Res.*, vol. 189, 2020.

[7] S. H. Low, "Convex relaxation of optimal power flow–part I: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.* vol. 1, no. 1, pp. 15–27, Mar. 2014.

[8] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput.*, 2015, pp. 402–409.

[9] A. Bernstein and E. Dall'Anese, "Linear power-flow models in multiphase distribution networks," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Europe*, 2017, pp. 1–6.

[10] M. B. Cain, R. P. O'neill, and A. Castillo, "History of optimal power flow and formulations," *Federal Energy Regulatory Commission* vol. 1, pp. 1–36, 2012.

[11] A.von Meier, *Electric Power Systems: A. Conceptual Introduction*. Hoboken, NJ, USA: Wiley, 2006.

[12] K. Baker, "Solutions of DC OPF are never AC feasible," in *Proc. 12th ACM Int. Conf. Future Energy Syst.*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 264–268.

[13] FERC, "Recent ISO software enhancements and future software and modeling plans," Accessed: Sep. 14, 2021. [Online]. Available: https://cms.ferc.gov/sites/default/files/2020-05/rto-iso-soft-2011.pdf

[14] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "A warm-start interior-point method for predictive control," in *Proc. UKACC Int. Conf. Control*, 2010, pp. 1–6.

[15] Q. Zhou, L. Tesfatsion, and C.-C. Liu, "Short-term congestion forecasting in wholesale power markets," *IEEE Trans. Power Syst.* vol. 26 no. 4, pp. 2185–2196, Nov. 2011.

[16] L. A. Roald and D. K. Molzahn, "Implied constraint satisfaction in power system optimization: The impacts of load variations," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.*, 2019, pp. 308–315.

[17] X. Ma, H. Song, M. Hong, J. Wan, Y. Chen, and E. Zak, "The security-constrained commitment and dispatch for midwest iso day-ahead co-optimized energy and ancillary service market," in *IEEE Power Energy Soc. Gen. Meeting*, 2009, pp. 1–8.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* vol. 521, pp. 436–442015.

[19] N. Guha, Z. Wang, M. Wytock, and A. Majumdar, "Machine learning for AC optimal power flow," 2019, *arXiv:1910.08842*.

[20] F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, "Predicting Ac optimal power flows: Combining deep learning and lagrangian dual methods," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 630–637.

[21] K. Baker, "Learning warm-start points for ac optimal power flow," *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process.*, 2019, pp. 1–6.

[22] A. Robson, M. Jamei, C. Ududec, and L. Mones, "Learning an optimally reduced formulation of Opf through meta-optimization," 2019, *arXiv:1911.06784*.

[23] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," *INFORMS J. Comput.* vol. 34, no. 1, pp. 463–480, Winter 2022.

[24] D. Deka and S. Misra, "Learning for Dc-Opf: Classifying active sets using neural nets," in *Proc. IEEE Milan PowerTech*, 2019, pp. 1–6.

[25] L. Chen and J. E. Tate, "Hot-starting the AC power flow with convolutional neural networks," 2020, *arXiv:2004.09342*.

[26] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5930–5934.

[27] T. Falconer and L. Mones, "Deep learning architectures for inference of Ac-Opf solutions," *NeurIPS Workshop Tackling Climate Change Mach. Learn.*, 2020.

[28] L. Halilbašić, F. Thams, A. Venzke, S. Chatzivasileiadis, and P. Pinson, "Data-driven security-constrained AC-OPF for operations and markets," in *Proc. Power Syst. Computation Conf.*, 2018, pp. 1–7.

[29] X. Pan, T. Zhao, M. Chen, and S. Zhang, "Deepopf: A. deep neural network approach for security-constrained DC optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1725–1735, May. 2021.

[30] F. Zhou, J. Anderson, and S. H. Low, "The optimal power flow operator: Theory and computation," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 1010–1022, Jun. 2021.

[31] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *Proc. IEEE Int. Conf. Commun., Control, Computing Technol. Smart Grids*, 2020, pp. 1–6.

[32] M. Jamei, L. Mones, A. Robson, L. White, J. Requeima, and C. Ududec, "Meta-optimization of optimal power flow," in *Proc. Int. Conf. Mach. Learn.*, 2019. [Online]. Availabe: https://www.climatechange.ai/papers/icml2019/42

[33] University of Washington: Department of Electrical & Computer Engineering, "Power systems test case archive," Accessed: Sep. 19, 2021. [Online]. Available: http://labs.ece.uw.edu/pstca/

[34] N. Dehmamy, A.-L. Barabási, and R. Yu, "Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15413–15423.

[35] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.* vol. 32 no. 1, pp. 4–24, Jan. 2021.

[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, 2017.

[38] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[39] C. Morris *et al.*, "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 01, pp. 4602–4609.

[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2017, *arXiv:1710.10903*.

[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.,* 32 no. 1, pp. 4–24, Jan. 2021 .

[42] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 869–877.

[43] M. Fey and J. E. Lenssen, "Fast graph representation learning with Pytorch geometric," *ICLR Workshop Representation Learn. Graphs Manifolds*, 2019.

[44] S. Babaeinejadsarookolaee *et al.*, "The power grid library for benchmarking Ac optimal power flow algorithms," 2019, *arXiv:1908.02788*.

[45] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.Jl: An open-source framework for exploring power flow formulations," in *Proc. Power Syst. Comput. Conf.*, 2018, pp. 1–8.

[46] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," 2012, *arXiv:1209.5145*.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[48] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[49] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[50] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[51] N. G. Singhal, J. Kwon, and K. W. Hedman, "Generator contingency modeling in electric energy markets: Derivation of prices via duality theory," 2019, *arXiv:1910.02323*.

[52] S. Liu, C. Wu, and H. Zhu, "Graph neural networks for learning real-time prices in electricity market," *ICML Workshop Tackling Climate Change Mach. Learn.*, 2021.

Thomas Falconer received the M.Sc. degree in energy systems and data analytics from the University College London, London, U. K., in 2020. He is currently working toward the Ph.D. degree with the Energy Markets and Analytics Section, Department of Wind and Energy Systems, Technical University of Denmark, Kongens Lyngby, Denmark. His research interests include machine learning, optimization, game theory, and the economics of data within a power systems context.

Letif Mones received the Ph.D. degree in theoretical and physical chemistry and structural chemistry from Eotvos Lorand University, Budapest, Hungary, in 2011. He was a Postdoctoral Research Associate with the Engineering Department of University of Cambridge, Cambridge, U.K., and with the Mathematics Institute of University of Warwick, Warwick, U.K. He is currently a Machine Learning Researcher with Invenia Labs, Cambridge, U.K. His research interests include application of machine learning techniques to infer solutions of optimal power flow and development of probabilistic models to forecast wholesale electricity prices.

Leveraging Power Grid Topology in Machine Learning Assisted Optimal Power Flow

Thomas Falconer and Letif Mones

***Abstract*—Machine learning assisted optimal power flow (OPF) aims to reduce the computational complexity of these non-linear and non-convex constrained optimization problems by consigning expensive (online) optimization to offline training. The majority of work in this area typically employs fully connected neural networks (FCNN). However, recently convolutional (CNN) and graph (GNN) neural networks have also been investigated, in effort to exploit topological information within the power grid. Although promising results have been obtained, there lacks a systematic comparison between these architectures throughout literature. Accordingly, we introduce a concise framework for generalizing methods for machine learning assisted OPF and assess the performance of a variety of FCNN, CNN and GNN models for two fundamental approaches in this domain: regression (predicting optimal generator set-points) and classification (predicting the active set of constraints). For several synthetic power grids with interconnected utilities, we show that locality properties between feature and target variables are scarce and subsequently demonstrate marginal utility of applying CNN and GNN architectures compared to FCNN for a fixed grid topology. However, with variable topology (for instance, modeling transmission line contingency), GNN models are able to straightforwardly take the change of topological information into account and outperform both FCNN and CNN models.**

***Index Terms*—OPF, graph theory, neural networks.**

NOMENCLATURE

Functions and operators

Φ, Ψ OPF operators that map grid parameters to optimal values of the primal variables and both primal and dual variables, respectively.

F OPF function introduced to simplify notation of the related operator whereby only grid parameters vary.

f Objective function of a particular OPF problem.

l Loss function used to optimize neural network parameters, θ.

Sets

$\mathcal{A}$ Set of active inequality constraints (those satisfied with equality at the optimal point).

$\mathcal{C}^{\mathrm{E}}$, $\mathcal{C}^{\mathrm{I}}$ Full sets of equality and inequality constraints for a particular OPF problem, respectively.

$\mathcal{F}_{\Phi}$ Set of feasible points for the optimization variables.

$\mathcal{M}$ Full set of neural network models for which predictive performance is assessed.

$\mathcal{N}$, $\mathcal{E}$ Sets of nodes (vertices) and edges that define an undirected graph, $\mathbb{G}$, respectively.

$\mathcal{V}$ Set of violated inequality constraints associated with a vector of optimization variables, y.

Ω Abstract set representing the OPF operator domain.

σ Set of hyperparameters used to define neural network architectures.

θ Set of neural network parameters optimized during the model training process.

Variables

P_g, P_l Power injection and withdrawal for a particular generator and load, respectively (active power components).

V_m Bus voltage magnitude.

x Vector of grid parameters (e.g. active and reactive power components of loads).

y Vector of primal variables (e.g. voltage magnitudes and active power component of generator injections).

z Vector of dual variables (Lagrangian multipliers) of the associated equality and inequality constraints.

Z_{ij} Impedance of transmission line between bus i and bus j.

I. INTRODUCTION

OPTIMAL power flow (OPF) is an umbrella term for a family of constrained optimization problems that govern electricity market dynamics and facilitate effective planning and operation of modern power systems [1, p. 514]. Classical OPF (AC-OPF) formulates a non-linear and non-convex economic dispatch model, which minimizes the cost of generator scheduling subject to either (or both) operation and security constraints of the grid [2]. By virtue of competitive efficiency, optimal schedules are typically found using interior-point methods [3]. However, the required computation of the Hessian (second-order derivatives) of the Lagrangian at each optimization step renders a super-linear time complexity, thus large-scale systems can be prohibitively slow to solve.

This computational constraint gives rise to several challenges for independent system operators (ISOs): (1) variable inclusion

Manuscript received 3 October 2021; revised 2 March 2022 and 26 April 2022; accepted 19 June 2022. Paper no. TPWRS-01564-2021. *(Corresponding author: Thomas Falconer.)*

The authors are with the Invenia Labs, 95 Regent Street, CB2 1AW Cambridge, U.K. (e-mail: thomas.falconer@invenialabs.co.uk; letif.mones@invenialabs.co.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TPWRS.2022.3187218.

Digital Object Identifier 10.1109/TPWRS.2022.3187218

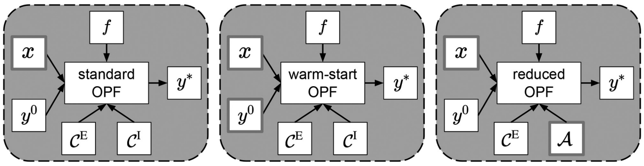


Fig. 1. Strategies for solving OPF with interior-point methods: standard (left), warm-start (center) and reduced (right) problems. x and y are the vectors of grid parameters and optimization variables, respectively, f is the objective function, $\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$ denote the sets of equality and inequality constraints, and $\mathcal{A} \subseteq \mathcal{C}^{\mathrm{I}}$ is the active subset of the inequality constraints. Typical varying arguments are highlighted in orange, whilst remaining arguments are potentially fixed.

of certain generators (i.e. unit commitment) invokes binary variables in the optimization model, thereby forming a mixed-integer, non-linear program (known to be NP-hard), exacerbating computational costs [4]; (2) the standard requirement for operators to satisfy N$-$1 security constraints (i.e. account for all contingency events where a single grid component fails) renders a much larger-scale problem, increasing the time complexity even further [5]; and lastly (3) modeling uncertainty in the supply-demand equilibrium induced by stochastic renewable generation requires methods such as scenario based Monte-Carlo simulation [6], which necessitates sequential OPF solutions at rates unattainable by conventional algorithms.

To overcome these challenges, ISOs often resort to simplified OPF models by utilizing convex relaxations [7] or linearizations [8], [9] such as the widely adopted DC-OPF model [10]. With considerably less control variables and constraints, DC-OPF can be solved very efficiently using interior-point or simplex methods [11, p. 224]. However, as DC-OPF solutions are in fact never feasible with respect to the full problem [12], set-points need to be found iteratively by manually updating the solution until convergence [13, p. 14] – hence DC-OPF is predisposed to sub-optimal generator scheduling.

In practice, ISOs typically leverage additional information about the grid in attempt to obtain solutions more efficiently. For instance, given the (reasonable) assumption that comparable grid states will correspond to neighbouring points in solution space, one can use the known solution to a similar problem as the starting value for the optimization variables of another problem – a so-called *warm-start* (Fig. 1, center panel) –, rendering considerably faster convergence compared to arbitrary initialisation [14]. Alternatively, ISOs can capitalize on the observation that only a fraction of inequality constraints are actually binding at the optimal point [15], hence one can remove a large number of constraints from the mathematical model and formulate an equivalent, but significantly cheaper, *reduced problem* [16] (Fig. 1, right panel). Security risks associated with the omission of violated constraints from the reduced problem can be mitigated by iteratively solving the reduced OPF and updating the active set until all constraints of the full problem are satisfied [17].

A. Machine Learning Assisted OPF

A compelling new area of research borne from the machine learning community attempts to alleviate reliance on subpar OPF

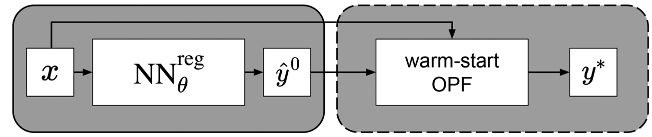


Fig. 2. Flowchart of the warm-start method (green panel) combined with a NN regressor (orange panel). For clarity, default arguments of the OPF operator are omitted.

frameworks by fitting an estimator functions on historical data. The estimators are typically neural networks (NNs) owed to their demonstrated ability to model complex non-linear relationships with negligible online computation [18]. This makes it possible to obtain predictions in real-time, thereby shifting the computational expense from online optimization to offline training – and the trained model can remain sufficient for a period of time, requiring only occasional re-training.

Most of the NN-based methods for machine learning assisted OPF can be generalized as one of two approaches: 1) *end-to-end* (or direct) models, where an estimator function is used to learn a direct mapping between the grid parameters and the optimal OPF solution; and 2) *hybrid* (or indirect) techniques – a two-step approach whereby an estimator function first maps the grid parameters to some quantities, which are subsequently used as inputs to an optimization problem to find a (possibly exact) solution. Based on the actual target type, these methods can be further categorized depending on the type of predicted quantity: regression or classification.

1) Regression: By inferring OPF solutions directly, end-to-end regression methods bypass conventional solvers altogether, offering the greatest (online) computational gains [19]. However, since OPF is a constrained optimization problem, the optimal solution is not necessarily a smooth function of the inputs: changes of the binding status of constraints can lead to abrupt changes of the optimal solution. Since the number of unique sets of binding constraints increases exponentially with system size, this approach requires training on relatively large data sets in order to obtain sufficient accuracy [20]. Moreover, there is no guarantee that the inferred solution is feasible, and violation of important constraints poses severe security risks to the grid.

Instead, one can adopt a hybrid approach whereby the inferred solution of the end-to-end method is used to initialize an interior-point solver (i.e. a warm-start), which provides an optimal solution to an optimization problem equivalent to the original one (Fig. 2). Compared to default heuristics used in the conventional optimization method, an accurate initial point could theoretically reduce the number of required iterations (and so the computational cost) to reach the optimal point [21]. However, as discussed in [22], there are several practical issues which could arise, leading to limited computational gain for this technique.

2) Classification: An alternative hybrid approach leverages the aforementioned technique of formulating a reduced problem by removing non-binding inequality constraints from the mathematical model. A NN classifier is first used to predict the active set of constraints by either 1) identifying all distinct active

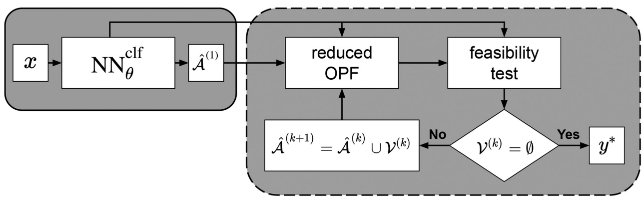


Fig. 3. Flowchart of the iterative feasibility test method (green panel) combined with a NN classifier (orange panel). $\hat{\mathcal{A}}^{(k)}$ and $\mathcal{V}^{(k)}$ are the sets of predicted active and violated inequality constraints at the k-th step of the iterative feasibility test, respectively. For clarity, default arguments of the OPF operator are omitted.

sets in the training data and using a multi-class classifier to map the features accordingly [23]; or 2) by predicting the binding status of each inequality constraint using a binary multi-label classifier [22]. Since the number of active sets increases exponentially with system size [24], the latter approach may be computationally favourable for larger grids.

To alleviate the security risks associated with imperfect classification, an *iterative feasibility test* can be employed to reinstate violated constraints until convergence, as detailed in [22] (Fig. 3). Since the reduced OPF is much cheaper relative to the full problem, this approach can in theory be rather efficient.

B. Contributions

Both the end-to-end and hybrid techniques for machine learning assisted OPF benefit from NN architectures designed to maximize predictive performance. Related works typically employ a range of shallow to deep fully connected neural networks (FCNN). However, convolutional (CNN) [25] and graph (GNN) [26]–[27] neural networks have recently been investigated to exploit assumed locality properties within the respective power grid, i.e. whether the topology of the electricity network influences the correlation between inputs and outputs.

Building on this set of works, our contributions are as follows:

- We introduce a concise framework for generalizing end-to-end and hybrid methods for machine learning assisted OPF by characterising them as estimators of the corresponding OPF operator or function.
- We provide a systematic comparison between the aforementioned NN architectures for both the regression and classification approaches.
- We demonstrate the marginal utility of applying CNN and GNN architectures for *fixed topology* problems (i.e. varying grid parameters only for the same topology), hence recommend the application of FCNN models for such problems.
- We show that locality properties between grid parameters (features or inputs) and corresponding generator set-points (targets or outputs) – essential for efficient inductive bias in both CNN and GNN models – are weak, which explains the moderate performance of these models compared to FCNN.
- We also show that a similar weak locality applies between grid parameters and locational marginal prices (LMPs), indicating that the applicability of CNN and GNN architectures would face similar challenges if instead used to predict these derived market signals.
- We present a set of *varying topology* problems (i.e. when both grid parameters and network topology are varied), that demonstrate successful utilization of structure based inductive bias through superior predictive performance of GNN models relative to both CNN and FCNN models.

It should be noted that, although we address the requirement of accurate predictions for machine learning assisted OPF, feasibility and optimality concerns associated with end-to-end methods, as well as the computational limitation of hybrid methods, remains a challenge for future work.

II. Methodology

A. Problem Formulation

This work centers on the fundamental form of OPF, without consideration for unit commitment or security constraints (although machine learning assisted OPF can be readily extended to such cases [28], [29]). In general, OPF problems can be expressed using the following concise form of mathematical programming:

$$\begin{aligned} \min_{y} \quad & f(x,y) \\ \text{s. t.} \quad & c_i^{\mathrm{E}}(x,y) = 0 \quad i = 1,\dots,n \\ & c_j^{\mathrm{I}}(x,y) \geq 0 \quad j = 1,\dots,m \end{aligned} \tag{1}$$

where x and y are the vectors of grid parameters and optimization variables, respectively, $f(x,y)$ is the objective (or cost) function (parameterized by x), which is minimized with respect to y and subject to equality constraints $c_i^{\mathrm{E}}(x,y) \in \mathcal{C}^{\mathrm{E}}$ and inequality constraints $c_j^{\mathrm{I}}(x,y) \in \mathcal{C}^{\mathrm{I}}$. For convenience, we introduce $\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$, which denote the sets of equality and inequality constraints with corresponding cardinalities $n = |\mathcal{C}^{\mathrm{E}}|$ and $m = |\mathcal{C}^{\mathrm{I}}|$. For instance, in a simple economic dispatch problem (the focus of this work), x includes the active and reactive power components of loads, y is a vector of voltage magnitudes and active powers of generators and the objective function is a quadratic or piece-wise linear function of the (monotonically increasing) generator cost curves. Equality constraints include the power balance and power flow equations, whilst inequality constraints impose lower and upper bounds on certain quantities.

B. OPF Operators and Functions

By formulating the problem in such a manner as (1), one can view OPF as an operator, which maps the grid parameters (x) to the optimal value of the optimization variables (y^*) [30]. In order to introduce a consistent framework, we extend the operator arguments by the objective (f) and constraint functions ($\mathcal{C}^{\mathrm{E}}$ and $\mathcal{C}^{\mathrm{I}}$), as well as by the starting value of the optimization variables (y^0). The value of y^0 has a considerable influence of the convergence rate of interior-point methods, and for non-convex formulations with multiple possible local minima, even the found optimum is a function of y^0. The general form of the

OPF operator can be written as[1]:

$$\Phi : \Omega \to \mathbb{R}^{n_y} : \quad \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) = y^*, \tag{2}$$

where Ω is an abstract set within which the values of the operator arguments are allowed to change and n_y denotes the dimension of the optimization variables. In the simplest case, only the grid parameters vary, whilst most arguments of the OPF operator remain fixed. Accordingly, we introduce a simpler notation, the OPF function, for such cases:

$$F_\Phi : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y} : \quad F_\Phi(x) = y^*, \tag{3}$$

where n_x and n_y are the dimensions of the grid parameters and optimization variables, respectively, whilst $\mathcal{F}_\Phi$ is used to denote the set of all feasible points, such that $y^* \in \mathcal{F}_\Phi$. Depending on the grid parameters, the problem may be infeasible: $\mathcal{F}_\Phi = \emptyset$.

C. Estimators of OPF Operators and Functions

Machine learning assisted OPF methods apply either an estimator operator or function, which both provide a computationally cheap prediction to the optimal point of the OPF based on the grid parameters, i.e. $\hat{\Phi}(x) = \hat{y}^* : \|\hat{y}^* - y^*\| < \varepsilon \wedge \mathbb{T}[\hat{\Phi}] \ll \mathbb{T}[\Phi]$ and $\hat{F}_\Phi(x) = \hat{y}^* : \|\hat{y}^* - y^*\| < \varepsilon \;\wedge\; \mathbb{T}[\hat{F}_\Phi] \ll \mathbb{T}[F_\Phi]$, where $\|\cdot\|$ is an arbitrary norm, ε is a threshold variable and $\mathbb{T}$ denotes the computational time to obtain the solution.

1) End-to-End: To learn the optimal OPF solution directly from the grid parameters, NNs as regressors can be used, depicted by the following function:

$$\hat{F}_\Phi(x) = \mathrm{NN}_\theta^{\mathrm{reg}}(x) = \hat{y}^*, \tag{4}$$

where subscript θ denotes the NN parameters and the superscript reg indicates that the NN is used as a regressor. The problem dimensionality can be reduced by predicting only a subset of the optimization variables – in this case, the remaining state variables can be easily obtained by solving the corresponding power flow problem [31], given the prediction is a feasible point. Optimal NN parameters can be obtained by minimizing some loss function between the ground-truth y^* and prediction $\hat{y}^*$ of some training set. Typically, the squared L2-norm, i.e. mean-squared error (MSE), is used: $\ell(y^*, \hat{y}^*) = \|y^* - \hat{y}^*\|_2^2$. To mitigate violations of certain constraints, a penalty term can be added to this loss function [20].

2) Warm-Start: Warm-start approaches utilize a hybrid model whereby a NN is first parameterized to infer an approximate set-point, $\hat{y}^0 = \mathrm{NN}_\theta^{\mathrm{reg}}(x)$, which is subsequently used to initialize the constrained optimization procedure resulting in the exact solution (y^*):

$$\begin{aligned} \hat{\Phi}^{\mathrm{warm}}(x) &= \Phi\left(x, \hat{y}^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) && (5)\\ &= \Phi\left(x, \mathrm{NN}_\theta^{\mathrm{reg}}(x), f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}\right) && (6)\\ &= y^*. && (7)\end{aligned}$$

[1]We note that an even more general form of the operator can be defined when the arguments are mapped to the joint space of the primal and dual variables of the optimization problem: $\Psi : \Omega \to \mathbb{R}^{n_y+n_z} : \quad \Psi(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}}) = (y^*, z^*)$, where z^* is the optimal value of the Lagrangian multipliers of the equality and inequality constraints. As locational marginal prices are computed from z^*, this formalism is useful to construct estimators for learning electricity prices.

Optimal NN parameters can be obtained by minimizing a similar conventional loss function as in the case of the end-to-end approach. However, significant improvement has been demonstrated by optimizing NN parameters with respect to a (meta-)loss function corresponding directly to the time complexity of the entire pipeline (i.e. including the warm-started OPF) [32]: $\ell(\hat{y}^0) = \mathbb{T}[\Phi(x, \hat{y}^0, f, \mathcal{C}^{\mathrm{E}}, \mathcal{C}^{\mathrm{I}})]$.

3) Reduced Problem: In this hybrid approach, a binary multi-label NN classifier ($\mathrm{NN}_\theta^{\mathrm{clf}}$) is used to predict the active set of constraints, and a reduced OPF problem is formulated, which maintains the same objective function as the original full problem:

$$\begin{aligned} \hat{\Phi}^{\mathrm{red}}(x) &= \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \hat{\mathcal{A}}\right) && (8)\\ &= \Phi\left(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \mathrm{NN}_\theta^{\mathrm{clf}}(x)\right) && (9)\\ &= \hat{y}^*, && (10)\end{aligned}$$

where $\mathcal{A} \subseteq \mathcal{C}^{\mathrm{I}}$ is the active subset of the inequality constraints and $\hat{\mathcal{A}}$ is the predicted active set. It should also be noted that $\mathcal{C}^{\mathrm{E}} \cup \mathcal{A}$ contains all active constraints defining the specific congestion regime. In the case of a multi-label classifier, the output is a binary vector representing an enumeration of the set of non-trivial constraints, learnt by minimizing the binary cross-entropy (BCE) loss between the ground-truths represented by $\mathcal{A}$ and the predicted binding probabilities of constraints defining $\hat{\mathcal{A}}$: $\ell(\mathcal{A}, \hat{\mathcal{A}}) = -\sum_j c_j \log \hat{c}_j + (1 - c_j)\log(1 - \hat{c}_j)$. The output dimension of the multi-label classifier is reduced by removing trivial constraints (those that are always binding or non-binding in the training set) for training. We note that to formulate the subsequent reduced OPF problem, these constraints need to be reinstated before the iterative feasibility test to construct the complete active set.

Violated constraints omitted from the reduced model are retained using the aforementioned iterative feasibility test to ensure convergence to an optimal point of the full problem. The computational gain can again be further enhanced via meta-optimization by directly encoding the time complexity into a (meta-)loss function and optimizing the NN weights accordingly [22]: $\ell(\hat{\mathcal{A}}) = \mathbb{T}[\Phi(x, y^0, f, \mathcal{C}^{\mathrm{E}}, \hat{\mathcal{A}})]$.

D. Architectures

Power grids are complex networks consisting of buses (e.g. generation points, load points etc.) connected by transmission lines, hence can conveniently be depicted as an un-directed graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denote the sets of nodes and edges (Fig. 4). Also, $\mathcal{G}$ and $\mathcal{L}$ will denote the sets of generators and loads, respectively.

This formulation motivates the use of NN architectures specifically designed to leverage the spatial dependencies within non-Euclidean data structures, i.e. GNN models – the hypothesis being that OPF problems exhibit a locality property whereby the network topology influences to correlation between grid parameters and the subsequent solution.

In real power grids, however, a given bus can include multiple generators and loads, which, although can have different power supply and demand, share the bus voltage. To accommodate such characteristics in GNN models straightforwardly, we use a

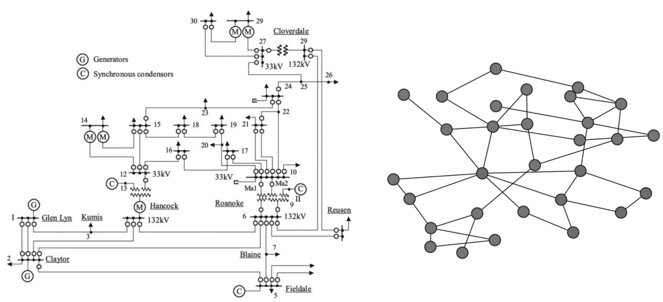


Fig. 4. Schematic diagram [33] (left) and corresponding graphical representation (right) for synthetic grid 30-ieee. Orange and green circles denote generator and load buses, respectively.

transformed version of the original graph: $\mathbb{G}' = (\mathcal{N}', \mathcal{E}')$, where each node of the transformed network represents either a single generator or a load (i.e. $|\mathcal{N}'| = |\mathcal{G}| + |\mathcal{L}|$), and generators and loads belonging to the same bus of the original network are interconnected. With this representation of the grid, generator real power outputs are obtained as individual nodal features, while bus voltage magnitudes are computed as averages of the corresponding individual voltages.

1) FCNN: Fully connected NN models, denoted by $\mathcal{M}^{\text{FCNN}}$, are used here as baseline. Their input domain is equivalent to the raw vector of grid parameters, i.e. active and reactive power components of loads: $x \in \mathbb{R}^{2|\mathcal{L}|}$, while the corresponding output vector includes the generators' injected active power and the voltage magnitude at buses comprising at least one generator ($\mathcal{N}^{\text{gen}} \in \mathcal{N}$), i.e. $y \in \mathbb{R}^{|\mathcal{G}|+|\mathcal{N}^{\text{gen}}|}$. Since FCNNs are defined in an un-structured data space, this baseline theoretically lacks sufficient relational inductive bias to efficiently exploit any underlying spatial dependencies – this information could be learnt implicitly through optimization, but possibly requires a highly flexible model with a large amount of data, thus scaling poorly to large-scale OPF problems [34]. We investigated two FCNN models using one ($\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$) and three ($\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$) hidden layers.

2) CNN: We explore the utility of augmenting the fully connected layers with an antecedent sequence of convolutional and pooling layers ($\mathcal{M}^{\text{CNN}}_{\text{global-4}}$), designed to extract a spatial hierarchy of latent features, which are subsequently (non-linearly) mapped to the target. A reasonable assumption here is that one can leverage spatial correlations within pseudo-images of the electrical grid using the weighted adjacency matrix. However, convolutions in Euclidean space are dependent upon particular geometric priors, which are not observed in the graph domain (e.g. shift-invariance), hence filters can no longer be node-agnostic and the lack of natural order means operations need to instead be permutation invariant. Nevertheless, we validate this conjecture using CNNs by combining each load constituent of length $|\mathcal{N}'|$ into a 3-dimensional tensor, i.e. $x \in \mathbb{R}^{2\times|\mathcal{N}'|\times|\mathcal{N}'|}$.

3) GNN: We analyze several GNN architectures whereby the weighted adjacency matrix is used to extract latent features by propagating information across neighbouring nodes irrespective of the input sequence [35]. Such propagation is achieved using graph convolutions, which can be broadly categorized as either spectral or spatial filtering [36].

Spectral filtering adopts methods from graph signal processing: operations occur in the Fourier domain whereby input signals are passed through parameterized functions of the normalized graph Laplacian, thereby exploiting its positive-semidefinite property. Given this procedure has $\mathcal{O}(|\mathcal{N}'|^3)$ time complexity, we investigate four spectral layers designed to reduce computational costs by avoiding full eigendecomposition of the Laplacian: (1) *ChebConv* ($\mathcal{M}^{\text{CHC}}$), which uses approximate filters derived from Chebyshev polynomials of the eigenvalues up to the K-th order [37]; (2) *GCNConv* ($\mathcal{M}^{\text{GCN}}$), which constrains the layer-wise convolution to first-order neighbours ($K = 1$), lessening overfitting to particular localities [38]; (3) *GraphConv* ($\mathcal{M}^{\text{GC}}$), which is analogous to *GCNConv* except adapting a discrete weight matrix for self-connections [39]; and (4) *GATConv* ($\mathcal{M}^{\text{GAT}}$), which extends the message passing framework of *GCNConv* by assigning each edge with relative importance through attention coefficients [40].

By contrast, spatial graph convolutions (a non-Euclidean generalization of the convolution operation found in CNNs) are performed directly in the graph domain, reducing the computational complexity whilst minimizing loss of structural information – a byproduct of reducing to embedded space [36]. We investigate *SplineConv* ($\mathcal{M}^{\text{SC}}$) [42] which, for a given node, computes a linear combination of its features together with those of its K-th order neighbours, weighted by a kernel function – the product of parameterized B-spline basis functions. The local support property of B-splines reduces the number of parameters, enhancing the computational efficiency of the operator. Note that all GNN models are named in accordance with the `PyTorch Geometric` library [43].

Finally, we note that due to the lack of connectivity information of the grid, conventional FCNN (and CNN) architectures typically fail to adapt efficiently to power system restructuring. In order to obtain sufficient performance with alternative grid topologies (i.e. contingency cases), these models need to be re-trained with appropriate training data. In contrast, GNNs compute localized convolutions in a manner such that the number of weights remains independent of the topology of the network making these models capable to train and predict on samples having different topologies [36].

E. Technical Details

1) Samples: To span multiple grid sizes, we built test cases using several synthetic grids from the Power Grid Library [44] ranging from 24 – 2853 buses. To maintain validity of the constructed data sets whilst ensuring a thorough exploration of congestion regimes, we generated 10 k (feasible) fixed topology samples for each synthetic grid by re-scaling each active and reactive load component (relative to nominal values) by factors independently drawn from a uniform distribution, $\mathcal{U}(0.8, 1.2)$. To investigate performance of the different NN architectures with varying topology, we also generated 10 k (feasible) samples subject to N$-$1 line contingency. For each sample, active and

TABLE I
NUMBER OF CHANNELS USED FOR CNN AND GNN ARCHITECTURES. σ_s AND σ_m ARE THE GRID SIZE AND MODEL TYPE BASED SCALING FACTORS. n_n DENOTES THE NUMBER OF NODES OF THE TRANSFORMED NETWORK AND n_y IS THE NUMBER OF OUTPUT VARIABLES

$$\sigma_s = \begin{cases} 1 & \text{if } |\mathcal{N}| \leq 73 \\ 2 & \text{if } |\mathcal{N}| > 73 \end{cases} \quad \sigma_m = \begin{cases} 1 & \text{if } \mathcal{M} = \mathcal{M}^{\text{GCN}} \text{ or } \mathcal{M}^{\text{GAT}} \\ 0.5 & \text{if } \mathcal{M} = \mathcal{M}^{\text{CHC}} \end{cases}$$

GNN layer	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{GNN}}_{\text{local-3}}$	$\mathcal{M}^{\text{GNN}}_{\text{global-4}}$
1.	4	$8\sigma_s$	8	8
2.	8	$16\sigma_s$	$n_n\sigma_m$	$n_n\sigma_m$
3.	16	—	$n_y\sigma_s\sigma_m$	$n_y\sigma_s\sigma_m$
Readout layer	yes	yes	no	yes

reactive load components were re-scaled as before and a single transmission line was randomly removed from the original grid topology. OPF solutions were obtained using `PowerModels.jl` [45] (an OPF package written in Julia [46]) in combination with the `IPOPT` solver [3].

2) Neural Networks: Our model with the largest number of parameters was the three hidden layer fully connected model ($\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$) that also served as the baseline. The size of each hidden layer was computed through a linear interpolation between the corresponding input and output sizes.

In the case of CNN, each model was constructed using 3×1 kernels, 1-dimensional max-pooling layers, zero-padding and a stride length of 1.

For GNN models, we investigated three architecture types: (1) the first type included two convolutional layers followed by a fully connected readout layer making the original local structure non-local ($\mathcal{M}^{\text{GNN}}_{\text{global-3}}$); (2) in the second type, only three convolutional layers were present, simply treating the features available locally at each node as the output ($\mathcal{M}^{\text{GNN}}_{\text{local-3}}$); and lastly (3) the third type was again a global one extending the above local type with a fully connected readout layer ($\mathcal{M}^{\text{GNN}}_{\text{global-4}}$). While corresponding $\mathcal{M}^{\text{GNN}}_{\text{global-3}}$ and $\mathcal{M}^{\text{GNN}}_{\text{local-3}}$ models were constructed to have an approximately equal number of parameters (details discussed below), $\mathcal{M}^{\text{GNN}}_{\text{global-4}}$ models had a significantly larger number of parameters due to the additional readout layer. For $\mathcal{M}^{\text{CHC}}$ and $\mathcal{M}^{\text{SC}}$ models, the hyperparameter K was set to 4.

Since our aim was to compare the predictive performance of models with and without topology based inductive bias, the single-layer FCNN, CNN and several GNN architectures were constructed to have a similar number of parameters for each synthetic grid. This required scaling the number of channels of the hidden layers of some architectures according to both the grid size (σ_s) and the model type (σ_m). We applied a simple grid search in order to obtain the optimal number of layers, as well as the values of parameters σ_s and σ_m. The actual number of channels used for the CNN and GNN models is presented in Table I.

Edge weights (e_{ij}) of the GNN architectures were modeled as a function of transmission line impedance, Z_{ij}, between the i-th and j-th bus. Specifically, we used the following general expression between connected buses i and j:

$$e_{ij} = \exp(-k \log |Z_{ij}|), \tag{11}$$

where k is a hyperparameter. Note that $k = 0$ leads to the application of the simple binary adjacency matrix, while in the case of $k = 1$ the absolute value of the corresponding element of the nodal admittance matrix is used.

For each grid, the generated 10 k samples were split into training, validation and test sets with a ratio of 80:10:10. In all cases, the ADAM [47] optimizer was applied (with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and learning-rate $\eta = 10^{-4}$) using an early stopping with a patience of 20 determined on the validation set. Mini-batch size of 100 was applied and hidden layers were equipped with BatchNorm [48] and a ReLU [49] activation function was used. For each model, statistics (mean and two-sided 95% confidence interval) of the predictive performance were computed using 10 independent runs.

Models were implemented in Python 3.0 using `PyTorch` [50] and `PyTorch Geometric` [43] libraries. Experiments were carried out on NVIDIA Tesla M60 GPUs. In order to facilitate research reproducibility in the field, we have made the generated samples, as well as the code our work is based upon, publicly available at `https://github.com/tdfalc/MLOPF.jl`.

III. NUMERICAL RESULTS

A. *Computational Performance of Prediction*

The fundamental motivation for using NN models to predict OPF solutions is their superior (online) computational performance compared to directly solving the corresponding AC-OPF problems. In Table II, we compared the average computational times of obtaining *exact* AC-OPF solutions using the `IPOPT` solver against inferring approximate solutions using various NN architectures. It is evident that, for all investigated systems, the computational time of the NN models is several orders of magnitude smaller than that of solving AC-OPF with conventional methods (note that in Table II, solve times of AC-OPF refer to a single sample, while prediction times of NN models refer to 1000 samples). Constrained optimization problems were solved on CPU (Intel Xeon E5-2686 v4, 2.3 GHz), while for the NN predictions we could utilize GPU (NVIDIA Tesla M60).

However, as discussed previously, comparing these computational times alone can be misleading: NN predictions are not necessarily optimal or even feasible. There have been several attempts to obtain feasible and possibly optimal estimates of OPF solutions (for instance by using hybrid approaches [29], [31] or introducing penalty terms of constraint violations in the loss function [20]). For all approaches, improving the quality of the predictive performance is fundamental. One apparent way is to increase the training data size significantly. In the following, we investigate the applicability of a more economical approach by using appropriate inductive bias in NN models.

TABLE II
PREDICTION TIME STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) FOR GLOBAL REGRESSION MODELS

Case	Solve time (ms)	Prediction time per 1000 samples (ms)						
	AC-OPF (IPOPT)	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	85.41 ± 1.04	10.86 ± 0.13	20.19 ± 0.11	191.75 ± 0.37	251.64 ± 2.99	196.64 ± 3.83	191.72 ± 0.75	236.52 ± 42.51
30-ieee	75.33 ± 0.63	10.58 ± 0.08	20.07 ± 0.11	194.45 ± 0.48	254.36 ± 2.16	197.59 ± 3.95	193.64 ± 0.88	237.33 ± 42.54
39-epri	147.47 ± 1.28	11.31 ± 0.14	21.47 ± 0.14	203.67 ± 1.99	269.31 ± 1.91	208.38 ± 4.43	204.75 ± 2.46	248.08 ± 43.68
57-ieee	125.24 ± 1.16	11.36 ± 0.06	21.06 ± 0.25	196.32 ± 0.27	257.18 ± 3.76	200.19 ± 4.55	196.12 ± 0.91	238.82 ± 41.02
73-ieee-rts	304.64 ± 1.32	13.25 ± 0.18	23.09 ± 0.41	216.67 ± 3.93	285.72 ± 7.28	220.83 ± 7.14	214.43 ± 3.25	260.34 ± 39.12
118-ieee	481.39 ± 2.68	12.59 ± 0.08	23.64 ± 1.94	200.02 ± 0.31	267.59 ± 3.88	203.14 ± 3.68	198.96 ± 0.29	245.38 ± 39.21
162-ieee-dtc	815.66 ± 6.27	13.81 ± 0.17	25.62 ± 2.39	207.86 ± 3.52	285.46 ± 7.57	215.16 ± 6.97	205.53 ± 3.72	261.93 ± 44.27
300-ieee	1467.43 ± 9.47	16.36 ± 0.08	28.04 ± 2.03	206.19 ± 0.74	301.14 ± 4.46	240.13 ± 3.28	203.19 ± 0.89	279.32 ± 42.01
588-sdet	2826.53 ± 51.2	22.03 ± 0.24	34.43 ± 2.36	240.94 ± 1.04	422.67 ± 3.56	363.13 ± 5.57	235.18 ± 0.64	354.07 ± 41.78
1354-pegase	10814.92 ± 29.6	36.04 ± 0.63	52.15 ± 8.89	390.56 ± 6.59	751.89 ± 9.86	676.29 ± 7.58	413.22 ± 4.71	520.68 ± 79.02
2853-sdet	34136.73 ± 99.1	76.54 ± 1.55	98.42 ± 2.42	1092.19 ± 5.79	1729.84 ± 8.66	1520.66 ± 9.32	1116.61 ± 9.94	1246.24 ± 39.39

TABLE III
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL REGRESSION MODELS (FIXED TOPOLOGY)

Case	MSE ($\times 10^{-3}$)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	0.18 ± 0.02	0.94 ± 0.04	1.55 ± 0.21	2.65 ± 0.13	$\mathbf{0.70 \pm 0.04}$	1.10 ± 0.12	1.04 ± 0.06	2.76 ± 0.19
30-ieee	0.05 ± 0.01	$\mathbf{0.03 \pm 0.01}$	0.62 ± 0.22	3.25 ± 0.82	0.09 ± 0.01	0.27 ± 0.08	0.26 ± 0.12	3.06 ± 0.33
39-epri	0.89 ± 0.10	3.16 ± 0.09	7.01 ± 0.09	4.30 ± 0.23	$\mathbf{2.38 \pm 0.10}$	3.00 ± 0.09	2.74 ± 0.13	4.72 ± 0.35
57-ieee	0.52 ± 0.11	1.62 ± 0.15	$\mathbf{1.22 \pm 0.10}$	2.18 ± 0.13	1.28 ± 0.14	1.64 ± 0.14	1.59 ± 0.14	2.28 ± 0.13
73-ieee-rts	0.21 ± 0.07	0.69 ± 0.02	1.06 ± 0.13	1.59 ± 0.11	$\mathbf{0.65 \pm 0.05}$	0.85 ± 0.11	0.85 ± 0.07	1.85 ± 0.21
118-ieee	0.39 ± 0.03	1.28 ± 0.07	3.68 ± 0.75	2.39 ± 0.12	$\mathbf{1.23 \pm 0.07}$	1.24 ± 0.07	1.27 ± 0.13	2.50 ± 0.10
162-ieee-dtc	2.61 ± 0.10	3.19 ± 0.08	3.28 ± 0.15	4.77 ± 0.21	3.08 ± 0.10	$\mathbf{2.90 \pm 0.11}$	3.04 ± 0.10	4.87 ± 0.23
300-ieee	2.06 ± 0.06	2.86 ± 0.05	3.95 ± 0.22	3.24 ± 0.09	2.42 ± 0.04	2.47 ± 0.20	$\mathbf{2.39 \pm 0.06}$	3.56 ± 0.19
588-sdet	2.56 ± 0.06	3.12 ± 0.05	4.10 ± 0.20	4.62 ± 0.36	3.25 ± 0.07	$\mathbf{3.00 \pm 0.06}$	3.05 ± 0.05	5.07 ± 0.30
1354-pegase	0.83 ± 0.12	$\mathbf{1.30 \pm 0.09}$	2.78 ± 0.23	2.16 ± 0.17	1.43 ± 0.09	1.35 ± 0.10	1.35 ± 0.12	2.51 ± 0.15
2853-sdet	5.99 ± 0.16	$\mathbf{6.87 \pm 0.05}$	15.71 ± 0.93	10.15 ± 0.58	9.70 ± 0.33	8.64 ± 0.29	8.49 ± 0.41	11.01 ± 0.46

TABLE IV
NUMBER OF PARAMETERS FOR GLOBAL REGRESSION MODELS (FIXED AND VARYING TOPOLOGY)

Case	# of parameters							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	6575	2156	1336	2303	2783	2943	2463	2353
30-ieee	4436	732	984	607	1087	1247	767	657
39-epri	7877	1580	1568	1035	1515	1675	1195	1085
57-ieee	13933	1610	1722	1047	1527	1687	1207	1097
73-ieee-rts	58677	19404	15504	18715	19195	19355	18875	18765
118-ieee	91835	25596	23160	26354	28178	28786	26962	26454
162-ieee-dtc	104396	7800	7524	8558	10382	10990	9166	8658
300-ieee	440480	82938	78006	83696	85520	86128	84304	83796
588-sdet	1512583	207152	200700	212838	214662	215270	213446	212938
1354-pegase	8486627	1408680	1390548	1409438	1411262	1411870	1410046	1409538
2853-sdet	42568525	9233926	9166558	9299404	9301228	9301836	9300012	9299504

B. *Fixed Topology*

We begin our analysis by investigating the predictive performance of NN models trained (and tested) using data derived from power grids with a fixed topology. In these experiments, only the grid parameters were varied within the datasets, while all the grid connections were the same among the samples. In this setup, FCNN and CNN architectures are functions of the grid parameters only, i.e. for regression and classification approaches we have $\text{NN}^{\text{reg}}_{\theta}(x_i) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i) = \hat{\mathcal{A}}_i$, where x_i is the grid parameter vector of the i-th sample. For GNN models, besides the grid parameters, the grid topology is also passed: $\text{NN}^{\text{reg}}_{\theta}(x_i, \mathbb{G}) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i, \mathbb{G}) = \hat{\mathcal{A}}_i$, where $\mathbb{G}$ represents the (fixed) grid topology with corresponding edge weights.

1) Regression: For each grid, Table III summarizes the MSE statistics for regression model architectures that encode the targets as global variables. The first column includes the results of our baseline $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ model, which has the largest number of parameters (Table IV). In the presence of appropriate locality attributes, CNN and GNN models are expected to provide a comparable performance to $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ with a significantly smaller amount of parameters due to their topology based inductive bias.

In order to investigate the predictive performance with and without topological information, we first constructed global

TABLE V
TRAINING TIME STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) FOR GLOBAL REGRESSION MODELS

Case	Training time ($\times 10^2$ s)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	0.75 ± 0.14	3.40 ± 0.47	$\mathbf{1.60 \pm 0.31}$	12.06 ± 1.76	20.78 ± 1.97	10.80 ± 1.87	11.81 ± 1.67	15.56 ± 2.64
30-ieee	0.57 ± 0.05	$\mathbf{0.58 \pm 0.04}$	1.07 ± 0.22	9.75 ± 3.03	16.04 ± 1.96	9.22 ± 1.30	14.06 ± 3.55	22.30 ± 6.16
39-epri	0.58 ± 0.10	$\mathbf{0.82 \pm 0.05}$	0.83 ± 0.17	12.23 ± 1.70	16.36 ± 2.90	8.69 ± 1.17	9.70 ± 0.74	15.66 ± 3.34
57-ieee	0.33 ± 0.08	$\mathbf{0.67 \pm 0.03}$	1.13 ± 0.17	12.73 ± 1.83	12.39 ± 2.99	9.20 ± 2.33	11.93 ± 2.10	13.69 ± 2.22
73-ieee-rts	0.83 ± 0.15	2.79 ± 0.12	$\mathbf{1.64 \pm 0.21}$	12.36 ± 1.82	19.13 ± 2.19	10.07 ± 1.49	12.29 ± 1.92	16.36 ± 2.53
118-ieee	0.43 ± 0.09	1.98 ± 0.18	$\mathbf{1.66 \pm 0.28}$	17.80 ± 2.44	8.25 ± 0.96	7.10 ± 0.77	5.73 ± 0.53	20.62 ± 1.99
162-ieee-dtc	0.28 ± 0.04	1.32 ± 0.17	$\mathbf{1.08 \pm 0.23}$	14.13 ± 2.56	6.45 ± 0.83	8.49 ± 1.69	7.44 ± 1.29	12.19 ± 1.89
300-ieee	0.33 ± 0.02	$\mathbf{0.64 \pm 0.05}$	1.70 ± 0.27	14.74 ± 1.94	11.87 ± 1.01	13.25 ± 1.63	8.43 ± 1.23	16.91 ± 5.28
588-sdet	0.65 ± 0.15	$\mathbf{0.58 \pm 0.05}$	1.84 ± 0.40	23.74 ± 6.36	11.24 ± 1.26	15.54 ± 3.02	10.72 ± 1.63	22.61 ± 4.16
1354-pegase	1.81 ± 0.22	$\mathbf{1.13 \pm 0.11}$	1.52 ± 0.43	18.07 ± 3.34	22.55 ± 1.46	26.74 ± 5.08	13.74 ± 1.77	21.54 ± 2.84
2853-sdet	9.54 ± 0.44	1.37 ± 0.05	$\mathbf{0.54 \pm 0.02}$	14.72 ± 1.00	16.93 ± 0.88	24.35 ± 2.19	14.38 ± 1.24	17.29 ± 3.29

TABLE VI
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (FIXED TOPOLOGY)

Case	MSE ($\times 10^{-3}$)					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	73.93 ± 8.46	$\mathbf{27.03 \pm 0.36}$	63.69 ± 9.76	2.63 ± 0.12	$\mathbf{0.50 \pm 0.04}$	2.48 ± 0.12
30-ieee	29.83 ± 5.39	$\mathbf{0.23 \pm 0.05}$	19.45 ± 6.46	2.39 ± 0.12	$\mathbf{0.06 \pm 0.01}$	2.84 ± 0.13
39-epri	14.46 ± 2.84	$\mathbf{3.27 \pm 0.18}$	15.09 ± 2.92	2.81 ± 0.14	$\mathbf{2.11 \pm 0.07}$	3.24 ± 0.19
57-ieee	8.53 ± 3.65	$\mathbf{2.29 \pm 0.15}$	9.80 ± 4.50	2.14 ± 0.15	$\mathbf{1.09 \pm 0.17}$	2.35 ± 0.22
73-ieee-rts	36.85 ± 1.53	$\mathbf{31.69 \pm 0.11}$	53.01 ± 1.03	1.31 ± 0.14	$\mathbf{0.35 \pm 0.04}$	1.67 ± 0.13
118-ieee	31.57 ± 3.29	$\mathbf{6.47 \pm 0.20}$	39.85 ± 7.85	3.91 ± 0.09	$\mathbf{1.41 \pm 0.09}$	4.34 ± 0.27
162-ieee-dtc	11.71 ± 0.61	$\mathbf{6.27 \pm 0.18}$	11.81 ± 0.60	6.40 ± 0.12	$\mathbf{3.47 \pm 0.11}$	5.55 ± 0.14
300-ieee	16.79 ± 2.59	$\mathbf{9.35 \pm 0.15}$	46.63 ± 8.50	3.48 ± 0.08	$\mathbf{2.83 \pm 0.08}$	5.01 ± 1.34
588-sdet	19.98 ± 2.27	$\mathbf{16.30 \pm 0.24}$	22.48 ± 0.95	5.64 ± 0.18	$\mathbf{4.20 \pm 0.07}$	15.51 ± 2.25

FCNN ($\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$), CNN ($\mathcal{M}^{\text{CNN}}_{\text{global-4}}$) and GNN ($\mathcal{M}^{\text{GNN}}_{\text{global-3}}$) models in a manner such that they have a similar number of parameters for each grid (Table IV).

In general, the regression performance of the investigated models (including the baseline) has a week correlation with the system size. This indicates that other factors, for instance the actual number of active sets, can also play an important role (as observed previously in [22]).

Comparing the CNN and GNN models, we found that in most of the cases, GNN models outperform the CNN model. An interesting exception is case 57-ieee, where the CNN model appeared to perform best. However, we rather consider this as an anomalous case, where the reduced error could be attributed to the coincidental unearthing of structural information within the receptive fields when convolving over the pseudo-image of the grid.

Although GCN is the simplest GNN model we investigated, in general it performs similarly to the more sophisticated GAT model. Whilst CHC and SC models have similar performance, computational efficiencies with respect to the training times of CHC (Table V) allude to a better scaling to larger grids.

The most striking observation is that the single-layer FCNN model exhibits exceedingly comparable performance to the best GNN models. For several cases, the difference between the average MSE values of the best GNN model and the single-layer model is not statistically significant and for the two largest grids, FCNN even outperforms all GNN models. It is also worth mentioning that $\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$ has at least one order of magnitude shorter training times than the global GNN models (Table V). For many cases, the significantly larger $\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$ model had an even shorter training time than $\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$ due to the faster convergence.

The moderate performance of the global GNN models could be a result of the readout layer, which simply induces noise by arbitrarily mixing signals of nodes further away in the system. To investigate this possibility, we performed a set of experiments up to grid size of 588, this time with local architectures for the GCN, CHC and GAT models (left three columns of Table VI). Interestingly, although the number of parameters of these local models is comparable to that of the global models (Table VII), the observed performance of each of the three GNN models is considerably worse. This suggests that the main contribution to the predictive capacity actually stems from the readout layer and also indicates a potential lack of locality properties.

To further validate the above arguments, we investigated the effect of extending the local models with a readout layer, i.e. converting the local regression models to their global counterparts. We found that using the readout layer significantly improved the predictive performance for all cases (right three columns of Table VI).

One could argue that the improvement is due to the increased number of parameters, which did indeed approximately double (Table VII). However, comparing the performance of the two sets of global models, the difference seems to be marginal,

TABLE VII
NUMBER OF PARAMETERS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (FIXED AND VARYING TOPOLOGY)

Case	# of parameters					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	2796	3165	2996	6888	7257	7088
30-ieee	796	1045	900	1528	1777	1632
39-epri	1355	1629	1493	2935	3209	3073
57-ieee	1541	1935	1703	3151	3545	3313
73-ieee-rts	20583	21451	21145	57411	58279	57973
118-ieee	27912	28835	28600	53508	54431	54196
162-ieee-dtc	9844	10969	10284	17644	18769	18084
300-ieee	87526	89662	88698	170464	172600	171636
588-sdet	220332	224469	222260	432412	436549	434340

TABLE VIII
BCE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL CLASSIFICATION MODELS (FIXED TOPOLOGY)

Case	BCE ($\times 10^{-2}$)						
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$
24-ieee-rts	1.89 ± 0.10	3.58 ± 0.12	4.66 ± 0.52	6.93 ± 0.65	$\mathbf{3.14 \pm 0.18}$	3.52 ± 0.21	3.42 ± 0.33
30-ieee	1.71 ± 0.31	5.14 ± 0.65	4.00 ± 0.55	8.76 ± 1.24	$\mathbf{3.58 \pm 0.28}$	5.33 ± 1.21	4.98 ± 0.73
39-epri	3.61 ± 0.12	7.55 ± 0.21	13.84 ± 0.22	10.48 ± 0.31	$\mathbf{7.07 \pm 0.15}$	8.07 ± 0.26	7.60 ± 0.35
57-ieee	1.67 ± 0.14	2.51 ± 0.24	2.51 ± 0.29	2.81 ± 0.17	2.34 ± 0.18	2.24 ± 0.24	$\mathbf{2.12 \pm 0.18}$
73-ieee-rts	3.06 ± 0.14	4.34 ± 0.10	4.71 ± 0.25	6.28 ± 0.24	$\mathbf{3.34 \pm 0.11}$	4.26 ± 0.59	4.08 ± 0.89
118-ieee	4.51 ± 0.25	6.19 ± 0.21	8.29 ± 0.39	7.86 ± 0.32	4.65 ± 0.19	$\mathbf{4.35 \pm 0.21}$	4.40 ± 0.20
162-ieee-dtc	5.42 ± 0.29	6.27 ± 0.15	6.31 ± 0.34	8.32 ± 0.19	6.19 ± 0.18	$\mathbf{5.99 \pm 0.17}$	6.18 ± 0.18
300-ieee	9.32 ± 0.23	$\mathbf{8.43 \pm 0.14}$	10.97 ± 0.29	10.20 ± 0.33	8.86 ± 0.19	8.70 ± 0.16	8.65 ± 0.21
588-sdet	10.92 ± 0.22	$\mathbf{8.75 \pm 0.14}$	12.13 ± 0.45	12.14 ± 0.37	11.38 ± 0.21	11.46 ± 0.18	10.92 ± 0.14
1354-pegase	11.99 ± 0.18	$\mathbf{10.56 \pm 0.10}$	21.56 ± 0.98	17.14 ± 0.44	18.80 ± 0.32	18.43 ± 0.93	17.86 ± 0.60
2853-sdet	17.30 ± 0.36	$\mathbf{11.55 \pm 0.04}$	37.88 ± 1.59	28.58 ± 0.88	31.83 ± 0.33	30.37 ± 0.53	33.47 ± 0.61

highlighting again the utility of the fully connected component and confirming our suspicion of a lack of locality within this problem.

Finally, we also investigated the utility of using the nodal admittance matrix to express electrical distances within the power grid – i.e. setting $k = 1$ in (11) –, rather than the simple binary adjacency matrix ($k = 0$). For this inherently more sophisticated approach, the results were in fact fairly consistent to those with $k = 0$ (a table summarising the MSE statistics for such models can be found in the *Supplementary Materials*). This is again in accordance with our suspicion that locality between input and output variables for this set of problems is rather limited, hence even more sophisticated measures of distance still cannot improve the performance of the GNNs.

2) Classification: In principle, the binding status of constraints could be predicted as nodal and edge features within a GNN framework. However, based on our findings for the regression experiments (i.e. that the global strategy significantly outperforms the local one), we treated constraints only as global variables. Classification performance is reported in terms of statistics of BCE of the test set, again based on 10 independent runs (Table VIII). Additional tables concerning the number of parameters as well as the training time for each model can be found in the *Supplementary Materials*.

Here, the single-layer FCNN was observed to be even more dominant relative to the regression case. Interestingly, for larger grids, it even outperforms the three-layer FCNN, which could be suffering from over-fitting as a consequence of increased flexibility. In general, we reach a similar conclusion as in the global regression setting, whereby the performance enhancements of the GNN classifiers are marginal respective to their practicality and computational limitations. CHC and SC models perform similarly, but CHC remains the cheaper option with respect to the training time. Note that GAT was excluded from these experiments since it had already shown weak performance for the regression case relative to the other GNN models.

Although for brevity we only present the test set loss, we also note that we observed a greater precision than recall in virtually every instance. This implies that the BCE objective is more sensitive to false positives. In combination with the iterative feasibility test, which is more sensitive to false negative predictions, this can result in a significant increase in the computational cost of obtaining solutions [22]. In order to fix this misalignment, one could either use a weighted BCE (with appropriate weights for the corresponding terms) or a meta-loss objective function [22] [32].

C. Varying Topology

We now focus our analysis toward the predictive performance of NN models trained (and tested) using data derived from power

TABLE IX
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL REGRESSION MODELS WITH VARYING TOPOLOGY

Case	MSE ($\times 10^{-3}$)							
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-3}}$
24-ieee-rts	1.27 ± 0.18	1.62 ± 0.16	1.42 ± 0.17	1.91 ± 0.17	$\mathbf{0.99 \pm 0.08}$	1.42 ± 0.18	1.25 ± 0.12	1.65 ± 0.13
30-ieee	8.77 ± 0.22	8.39 ± 0.19	8.53 ± 0.18	8.68 ± 0.16	$\mathbf{0.23 \pm 0.04}$	1.92 ± 0.62	0.66 ± 0.08	3.43 ± 0.37
39-epri	12.72 ± 0.28	12.09 ± 0.22	13.33 ± 0.21	12.56 ± 0.24	$\mathbf{3.31 \pm 0.16}$	5.65 ± 0.85	4.23 ± 0.23	7.86 ± 0.33
57-ieee	4.34 ± 0.12	3.88 ± 0.13	4.01 ± 0.12	3.96 ± 0.13	$\mathbf{0.82 \pm 0.08}$	2.81 ± 0.72	1.27 ± 0.13	2.43 ± 0.16
73-ieee-rts	0.85 ± 0.05	0.95 ± 0.05	1.01 ± 0.04	1.16 ± 0.06	$\mathbf{0.66 \pm 0.07}$	0.92 ± 0.06	0.86 ± 0.07	1.24 ± 0.18
118-ieee	3.06 ± 0.14	2.59 ± 0.12	2.88 ± 0.11	2.86 ± 0.12	$\mathbf{1.15 \pm 0.0}5$	1.78 ± 0.12	1.38 ± 0.08	2.66 ± 0.34
162-ieee-dtc	5.37 ± 0.18	4.38 ± 0.17	4.59 ± 0.13	5.81 ± 0.15	4.27 ± 0.13	5.29 ± 0.14	$\mathbf{3.95 \pm 0.20}$	5.29 ± 0.16
300-ieee	3.24 ± 0.08	3.16 ± 0.08	4.02 ± 0.27	3.62 ± 0.09	$\mathbf{2.42 \pm 0.07}$	2.79 ± 0.07	2.64 ± 0.06	3.72 ± 0.08
588-sdet	4.95 ± 0.12	4.02 ± 0.12	4.98 ± 0.14	4.63 ± 0.14	3.83 ± 0.31	4.16 ± 0.13	$\mathbf{3.36 \pm 0.56}$	4.46 ± 0.13

TABLE X
MSE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR LOCAL AND EXTENDED GLOBAL REGRESSION GNN MODELS (VARYING TOPOLOGY)

Case	MSE ($\times 10^{-3}$)					
	$\mathcal{M}^{\text{GCN}}_{\text{local-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{local-3}}$	$\mathcal{M}^{\text{GAT}}_{\text{local-3}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-4}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-4}}$	$\mathcal{M}^{\text{GAT}}_{\text{global-4}}$
24-ieee-rts	76.18 ± 8.12	$\mathbf{26.59 \pm 0.42}$	71.32 ± 9.76	1.88 ± 0.16	$\mathbf{0.74 \pm 0.12}$	1.57 ± 0.11
30-ieee	13.35 ± 2.06	$\mathbf{3.41 \pm 0.08}$	16.55 ± 6.37	8.68 ± 0.18	$\mathbf{0.15 \pm 0.02}$	2.13 ± 0.37
39-epri	46.47 ± 8.69	$\mathbf{4.47 \pm 0.22}$	24.97 ± 7.03	12.51 ± 0.21	$\mathbf{2.97 \pm 0.13}$	6.75 ± 1.06
57-ieee	9.13 ± 2.97	$\mathbf{2.09 \pm 0.23}$	24.13 ± 9.47	4.11 ± 0.12	$\mathbf{0.66 \pm 0.08}$	1.74 ± 0.21
73-ieee-rts	70.21 ± 1.84	$\mathbf{65.43 \pm 0.06}$	99.92 ± 7.04	1.23 ± 0.17	$\mathbf{0.42 \pm 0.04}$	1.23 ± 0.26
118-ieee	22.55 ± 6.68	$\mathbf{4.88 \pm 0.05}$	35.95 ± 4.88	3.02 ± 0.12	$\mathbf{1.55 \pm 0.12}$	2.97 ± 0.34
162-ieee-dtc	24.74 ± 7.34	$\mathbf{6.48 \pm 0.34}$	19.65 ± 7.61	6.38 ± 0.12	$\mathbf{4.77 \pm 0.16}$	6.57 ± 1.08
300-ieee	16.86 ± 2.55	$\mathbf{6.84 \pm 0.19}$	50.52 ± 7.97	3.68 ± 0.09	$\mathbf{3.02 \pm 0.13}$	4.67 ± 0.92
588-sdet	11.87 ± 5.38	$\mathbf{7.18 \pm 0.18}$	22.62 ± 0.19	4.89 ± 0.14	$\mathbf{4.36 \pm 0.13}$	6.96 ± 0.98

grids of size 24 – 588 with varying topology. In these experiments, we modeled the N$-$1 line contingency and samples for a given grid differed not only in their input grid parameters but also in their topology. For FCNN and CNN models, we used only grid parameters as inputs to predict the corresponding quantities of regression and classification, similarly to the fixed topology. We note that in theory, the input vector could be extended to include topological information, but it is rather cumbersome due to the quadratic scaling of the weighted adjacency matrix with system size. For GNN models, however, the change in the topology can be naturally taken into account by passing the graph information of the sample along with the grid parameters. For the regression and classification approaches we have: $\text{NN}^{\text{reg}}_{\theta}(x_i, \mathbb{G}_i) = \hat{y}^*_i$ and $\text{NN}^{\text{clf}}_{\theta}(x_i, \mathbb{G}_i) = \hat{\mathcal{A}}_i$, where x_i and $\mathbb{G}_i$ are the grid parameter vector and topology of the i-th sample, respectively.

1) Regression: We begin our discussion again by evaluating the global regression models (Table IX). As expected, due to the larger effective parameter space, the regression performance using samples with varying topology decreases when compared to those with fixed topology for all cases and architectures (c.f. Table III). A significant difference is that the best GNN models – CHC in most cases – outperforms both the single-layer and even the three-layer FCNN models (and CNN models too). This is resultant of the fact that in these models, any change in the network topology is ignored, whilst in the GNN architectures it is considered explicitly. This is a promising finding for applications of GNN models for predicting solutions of more sophisticated OPF problems including contingencies.

Interestingly, further investigations revealed that locality properties still play a marginal role in the predictive performance of GNNs: as for the fixed topology cases, local GNN models have a significantly weaker performance, which is subsequently restored by attaching a readout layer (Table X).

2) Classification: For the classification models, we considered again only the global case (Table XI). We note that due to the higher number of non-trivial constraints, the size of the NN models with varying topology differs from those with fixed topology (details are shown in the *Supplementary Materials*). Therefore, unlike in the case of regression, we cannot compare directly the BCE statistics of experiments with fixed and varying topology. Nevertheless, in general, we found a similar trend to the global regression, i.e. the best performing GNN model (again, most often CHC) consistently outperforms the single-layer FCNN, the CNN and even the three-layer FCNN models. This means that applying GNN models is preferable over a significantly larger FCNN architecture for both OPF related regression and classification based problems with varying topology.

D. Locality Properties

Experimental results for the NN models indicated that the general assumption of locality may not be appropriate for this problem, i.e. there is only a weak – or no existence of – locality between load inputs and generator set-point outputs. To explore this relationship further, we carried out a sensitivity analysis that directly measures locality: for each synthetic grid, we iteratively

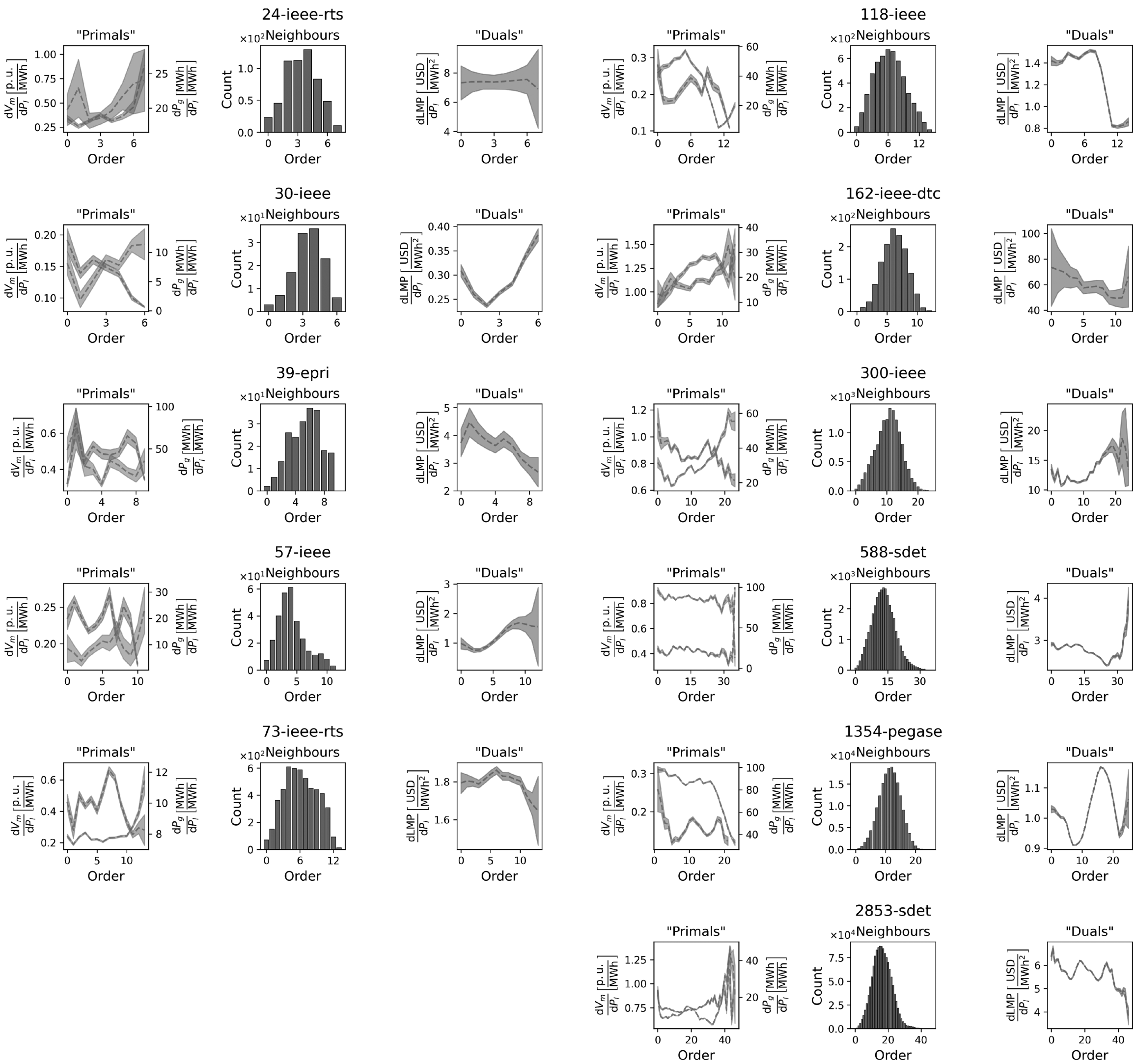


Fig. 5. Analysis of locality properties for each synthetic grid. Left and right panels show the average absolute value of the relative change (with two-sided 95% confidence intervals) in voltage magnitude (green), injected active power (orange) and locational marginal prices (purple), respectively, as a function of the topological distance from the perturbed load. Center panels show the histogram of generators with respect to the neighbourhood order from loads.

TABLE XI
BCE STATISTICS (MEAN AND TWO-SIDED 95% CONFIDENCE INTERVALS) OF THE TEST SETS FOR GLOBAL CLASSIFICATION MODELS WITH VARYING TOPOLOGY

Case	BCE ($\times 10^{-2}$)						
	$\mathcal{M}^{\text{FCNN}}_{\text{global-3}}$	$\mathcal{M}^{\text{FCNN}}_{\text{global-1}}$	$\mathcal{M}^{\text{CNN}}_{\text{global-4}}$	$\mathcal{M}^{\text{GCN}}_{\text{global-3}}$	$\mathcal{M}^{\text{CHC}}_{\text{global-3}}$	$\mathcal{M}^{\text{SC}}_{\text{global-3}}$	$\mathcal{M}^{\text{GC}}_{\text{global-3}}$
24-ieee-rts	3.56 ± 0.36	3.19 ± 0.17	3.32 ± 0.18	3.43 ± 0.16	$\mathbf{1.44 \pm 0.11}$	1.81 ± 0.16	1.67 ± 0.16
30-ieee	6.21 ± 0.35	6.19 ± 0.32	6.22 ± 0.33	6.59 ± 0.37	$\mathbf{3.03 \pm 0.18}$	4.81 ± 1.35	4.43 ± 0.17
39-epri	8.66 ± 0.19	8.51 ± 0.17	9.06 ± 0.21	8.78 ± 0.21	$\mathbf{3.74 \pm 0.12}$	5.71 ± 0.86	4.36 ± 0.19
57-ieee	5.34 ± 0.24	4.56 ± 0.17	4.65 ± 0.18	4.59 ± 0.15	$\mathbf{1.88 \pm 0.07}$	3.48 ± 0.93	2.17 ± 0.09
73-ieee-rts	3.98 ± 0.22	3.87 ± 0.16	3.69 ± 0.15	4.18 ± 0.28	$\mathbf{2.25 \pm 0.12}$	2.84 ± 0.22	2.92 ± 0.21
118-ieee	4.75 ± 0.15	3.95 ± 0.11	4.27 ± 0.14	4.28 ± 0.12	2.82 ± 0.06	3.42 ± 0.14	$\mathbf{2.79 \pm 0.12}$
162-ieee-dtc	3.19 ± 0.14	2.66 ± 0.06	2.71 ± 0.06	3.23 ± 0.06	$\mathbf{2.17 \pm 0.07}$	2.65 ± 0.15	2.66 ± 0.07
300-ieee	8.07 ± 0.17	7.35 ± 0.11	7.88 ± 0.14	7.43 ± 0.17	$\mathbf{6.38 \pm 0.14}$	6.79 ± 0.14	6.74 ± 0.17
588-sdet	6.84 ± 0.77	5.91 ± 0.07	6.16 ± 0.13	5.87 ± 0.12	$\mathbf{5.12 \pm 0.08}$	6.15 ± 0.11	5.91 ± 0.08

perturbed each active load of 100 configurations by 1% and recorded the absolute value of the relative change in voltage magnitude and active power injection of each generator (i.e. $|\frac{dV_m^j}{dP_l^i}|$ and $|\frac{dP_g^j}{dP_l^i}|$, where P_l^i are the active loads with $i = 1, \ldots, |\mathcal{L}|$; and V_m^j and P_g^j are the voltage magnitude and injected active power of generators with $j = 1, \ldots, |\mathcal{G}|$), as a function of neighbourhood order (i.e. the topological distance from the perturbed load). If a grid were to exhibit locality properties, one would expect a distinct negative correlation between the average of these quantities and the respective distance from the perturbed load within the graph domain.

The results of the sensitivity analysis are shown in the left panels of Fig. 5. Although there are certain cases where either the voltage magnitude or active power injection show a weak anti-correlation with the topological distance, in general we found little evidence that the topology of the network influences the correlation between input and output variables. Plotting the distribution of generators as a function of distance from the perturbed load (middle panels of Fig. 5) suggests that this result should be of no surprise: as the system size increases, so does the average distance between the perturbed load and the generators in the system, which decreases the likelihood that nearby generators will balance corresponding demand (for apparent physical reasons such as generator capacity, line congestion etc.).

Finally, we also explored the existence of possible locality between grid inputs and the LMPs, which are functions of the duals (shadow prices) [51]. If a stronger locality property were to exist here this would be promising for using GNN models to predict electricity prices even with fixed topology [52]. However, as shown in the right panels of Fig. 5, we found no evidence of locality for the LMP values either.

IV. Conclusion

With the potential to shift the entire computational effort to offline training, machine learning assisted OPF has become an increasingly interesting research direction. Neural network based approaches are particularly promising as they can effectively model complex non-linear relationships between grid parameters and primal or dual variables of the underlying OPF problem.

Although most related works have applied fully connected neural networks so far, these networks scale relatively poorly with system size. Therefore, incorporating topological information of the electricity grid into the inductive bias of some graph neural network is a sensible step towards reducing the number of NN parameters.

In this paper, we first provided a general framework of the most widely used end-to-end and hybrid techniques and showed that they can be considered as estimators of the OPF operator or function. In this sense, our framework could be readily extended to more sophisticated OPF problems, such as consideration of unit commitment or security constraints, as well as direct prediction of derived market signals (e.g. LMPs).

We then presented a systematic comparison of several NN architectures including FCNN, CNN and GNN models. We found that for systems with fixed topology, an FCNN model has a comparable or even better predictive performance than global CNN and GNN models with similar number of parameters. The moderate performance of the CNN model can be explained by the fact that it carries out convolutions in Euclidean space (instead of the graph domain). We also identified that in the case of global GNN models, the readout layer plays a key role: constructing local models by removing their readout layer led to a significant decline in the predictive performance.

The results with fixed topology indicated that the required assumption of locality between grid parameters (inputs) and generator set-points (outputs) might not hold. To validate the findings of the NN experiments, by carrying out a sensitivity analysis we showed that locality properties are indeed scarce between grid parameters and primal variables of the OPF. Additionally, we found a similar lack of locality between grid parameters and LMPs.

Finally, we also performed a systematic comparison of NN models using varying topology of the samples. In these experiments, we modeled the N$-$1 contingency of transmission lines in both the training and test sets. We found that for such cases, global GNN architectures outperform FCNN and CNN models for both regression and classification based problems. The reason is that although locality properties still play a limited role, GNN models could take the changes of the topology into account, which were completely neglected amongst FCNN and CNN models in our setup. Although it might be possible to extend FCNN and CNN models' input by topology related features, it is definitely less straightforward than for GNN models, where this information is accounted for naturally. This property of the GNN architectures therefore makes these models promising for realistic applications, especially for security constrained OPF problems.

References

[1] A. J. Wood, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2014.

[2] R. Billinton and W. Li, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Cham, Switzerland: Springer, 1994.

[3] A. Wachter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.* vol. 106, no. 1, pp. 25–57, 2006.

[4] A. Castillo, C. Laird, C. A. Silva-Monroy, J. Watson, and R. P. O'Neill, "The unit commitment problem with ac optimal power flow constraints," *IEEE Trans. Power Syst.* vol. 31, no. 6, pp. 4853–4866, Nov. 2016.

[5] J. Rahman, C. Feng, and J. Zhang, "Machine learning-aided security constrained optimal power flow," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2020, pp. 1–5.

[6] I. Mezghani, S. Misra, and D. Deka, "Stochastic Ac Optimal Power Flow: A Data-Driven Approach," *Elect. Power Syst. Res.*, vol. 189, 2020.

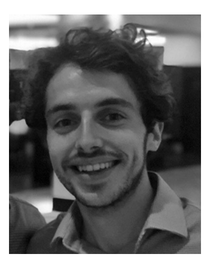
[7] S. H. Low, "Convex relaxation of optimal power flow–part I: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.* vol. 1, no. 1, pp. 15–27, Mar. 2014.

[8] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput.*, 2015, pp. 402–409.

[9] A. Bernstein and E. Dall'Anese, "Linear power-flow models in multiphase distribution networks," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Europe*, 2017, pp. 1–6.

[10] M. B. Cain, R. P. O'neill, and A. Castillo, "History of optimal power flow and formulations," *Federal Energy Regulatory Commission* vol. 1, pp. 1–36, 2012.

[11] A.von Meier, *Electric Power Systems: A. Conceptual Introduction*. Hoboken, NJ, USA: Wiley, 2006.
[12] K. Baker, "Solutions of DC OPF are never AC feasible," in *Proc. 12th ACM Int. Conf. Future Energy Syst.*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 264–268.
[13] FERC, "Recent ISO software enhancements and future software and modeling plans," Accessed: Sep. 14, 2021. [Online]. Available: https://cms.ferc.gov/sites/default/files/2020-05/rto-iso-soft-2011.pdf
[14] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "A warm-start interior-point method for predictive control," in *Proc. UKACC Int. Conf. Control*, 2010, pp. 1–6.
[15] Q. Zhou, L. Tesfatsion, and C.-C. Liu, "Short-term congestion forecasting in wholesale power markets," *IEEE Trans. Power Syst.* vol. 26 no. 4, pp. 2185–2196, Nov. 2011.
[16] L. A. Roald and D. K. Molzahn, "Implied constraint satisfaction in power system optimization: The impacts of load variations," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.*, 2019, pp. 308–315.
[17] X. Ma, H. Song, M. Hong, J. Wan, Y. Chen, and E. Zak, "The security-constrained commitment and dispatch for midwest iso day-ahead co-optimized energy and ancillary service market," in *IEEE Power Energy Soc. Gen. Meeting*, 2009, pp. 1–8.
[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* vol. 521, pp. 436–442015.
[19] N. Guha, Z. Wang, M. Wytock, and A. Majumdar, "Machine learning for AC optimal power flow," 2019, *arXiv:1910.08842*.
[20] F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, "Predicting Ac optimal power flows: Combining deep learning and lagrangian dual methods," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 630–637.
[21] K. Baker, "Learning warm-start points for ac optimal power flow," *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process.*, 2019, pp. 1–6.
[22] A. Robson, M. Jamei, C. Ududec, and L. Mones, "Learning an optimally reduced formulation of Opf through meta-optimization," 2019, *arXiv:1911.06784*.
[23] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," *INFORMS J. Comput.* vol. 34, no. 1, pp. 463–480, Winter 2022.
[24] D. Deka and S. Misra, "Learning for Dc-Opf: Classifying active sets using neural nets," in *Proc. IEEE Milan PowerTech*, 2019, pp. 1–6.
[25] L. Chen and J. E. Tate, "Hot-starting the AC power flow with convolutional neural networks," 2020, *arXiv:2004.09342*.
[26] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 5930–5934.
[27] T. Falconer and L. Mones, "Deep learning architectures for inference of Ac-Opf solutions," *NeurIPS Workshop Tackling Climate Change Mach. Learn.*, 2020.
[28] L. Halilbašić, F. Thams, A. Venzke, S. Chatzivasileiadis, and P. Pinson, "Data-driven security-constrained AC-OPF for operations and markets," in *Proc. Power Syst. Computation Conf.*, 2018, pp. 1–7.
[29] X. Pan, T. Zhao, M. Chen, and S. Zhang, "Deepopf: A. deep neural network approach for security-constrained DC optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1725–1735, May. 2021.
[30] F. Zhou, J. Anderson, and S. H. Low, "The optimal power flow operator: Theory and computation," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 1010–1022, Jun. 2021.
[31] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *Proc. IEEE Int. Conf. Commun., Control, Computing Technol. Smart Grids*, 2020, pp. 1–6.
[32] M. Jamei, L. Mones, A. Robson, L. White, J. Requeima, and C. Ududec, "Meta-optimization of optimal power flow," in *Proc. Int. Conf. Mach. Learn.*, 2019. [Online]. Availabe: https://www.climatechange.ai/papers/icml2019/42
[33] University of Washington: Department of Electrical & Computer Engineering, "Power systems test case archive," Accessed: Sep. 19, 2021. [Online]. Available: http://labs.ece.uw.edu/pstca/
[34] N. Dehmamy, A.-L. Barabási, and R. Yu, "Understanding the Representation Power of Graph Neural Networks in Learning Graph Topology," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15413–15423.
[35] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst* vol. 32 no. 1, pp. 4–24, Jan. 2021.
[37] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, 2017.
[38] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
[39] C. Morris *et al.*, "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 01, pp. 4602–4609.
[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2017, *arXiv:1710.10903*.
[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.,* 32 no. 1, pp. 4–24, Jan. 2021 .
[42] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous B-spline kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 869–877.
[43] M. Fey and J. E. Lenssen, "Fast graph representation learning with Pytorch geometric," *ICLR Workshop Representation Learn. Graphs Manifolds*, 2019.
[44] S. Babaeinejadsarookolaee *et al.*, "The power grid library for benchmarking Ac optimal power flow algorithms," 2019, *arXiv:1908.02788*.
[45] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.Jl: An open-source framework for exploring power flow formulations," in *Proc. Power Syst. Comput. Conf.*, 2018, pp. 1–8.
[46] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," 2012, *arXiv:1209.5145*.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[48] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[49] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.
[50] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[51] N. G. Singhal, J. Kwon, and K. W. Hedman, "Generator contingency modeling in electric energy markets: Derivation of prices via duality theory," 2019, *arXiv:1910.02323*.
[52] S. Liu, C. Wu, and H. Zhu, "Graph neural networks for learning real-time prices in electricity market," *ICML Workshop Tackling Climate Change Mach. Learn.*, 2021.



Thomas Falconer received the M.Sc. degree in energy systems and data analytics from the University College London, London, U. K., in 2020. He is currently working toward the Ph.D. degree with the Energy Markets and Analytics Section, Department of Wind and Energy Systems, Technical University of Denmark, Kongens Lyngby, Denmark. His research interests include machine learning, optimization, game theory, and the economics of data within a power systems context.

Letif Mones received the Ph.D. degree in theoretical and physical chemistry and structural chemistry from Eotvos Lorand University, Budapest, Hungary, in 2011. He was a Postdoctoral Research Associate with the Engineering Department of University of Cambridge, Cambridge, U.K., and with the Mathematics Institute of University of Warwick, Warwick, U.K. He is currently a Machine Learning Researcher with Invenia Labs, Cambridge, U.K. His research interests include application of machine learning techniques to infer solutions of optimal power flow and development of probabilistic models to forecast wholesale electricity prices.