Optimal Power Flow With Physics-Informed Typed Graph Neural Networks

Tania B. Lopez-Garcia and José Antonio Domínguez-Navarro, *Senior Member, IEEE*

***Abstract*—This work describes a new way to solve the optimal power flow problem applying typed graph neural networks. Typed graph neural networks allow the representation of different elements of transmission systems with different types of nodes, thus adding accuracy and interpretability to the solutions obtained, in comparison to results obtained with conventional feed-forward neural network models. The proposed graph neural network architecture is trained without the need of training data, through a physics informed loss function which incorporates not only the optimization objective, but also operational constraints of the physical system. Results are comparable with those obtained with the interior point method, and it is shown that the calculation time is greatly reduced.**

***Index Terms*—Optimal power flow, graph neural networks, unsupervised learning, physics-informed neural networks.**

I. Introduction

THE optimal power flow (OPF) problem is a fundamental optimization tool in power system operation and planning. Initially formulated by Carpentier in 1962 [1], extending the economic dispatch problem by incorporating power flow equations, but since then the OPF problem has undergone significant evolution in both modeling and algorithmic approaches. A comprehensive overview of the objective functions and algorithms used in classical electrical networks can be found in [2], [3], [4], [5], [6], [7].

However, in the early twenty-first century, the landscape shifted towards more intricate OPF models [8], [9], [10], addressing challenges in the electricity market [11], smart grids [12], and renewable energy integration [13]. These developments necessitate the use of probabilistic models [14], [15] and exploration of numerous scenarios [16] to capture uncertainties in renewable energy and market dynamics. The increasing complexity of models and operational flexibility of networks demand considerable time for study execution and require faster analysis [17].

In response to these challenges, several authors propose the application of neural networks and machine learning in OPF. Various articles explore the intersection of OPF and neural networks [16], [18], [19], leveraging these technologies to enhance the speed and efficiency of studies. Recent trends indicate a growing preference for machine learning in solving AC optimal power flow due to the significant runtime speedup compared to traditional optimization techniques. However, challenges arise, as some methods require large amounts of target data and struggle with generalization outside training scenarios [16].

Approaches taken with Physics-Informed Neural Networks (PINN) stem from the formulation of the problem, introducing constraints into the loss function guiding neural network learning. In [20], four stacked Multilayer Perceptrons (MLP) are used, one for each significant variable (voltage magnitude, phase angle, active power, and reactive power of generators), with constraints incorporated into the loss function. In [21], deep reinforcement learning is utilized, employing an action-value function mimicking the gradient ascent algorithm to update the action network. In [22], prior information provided by the Karush-Kuhn-Tucker conditions is used to train the network. In [23], information supplied by the Jacobian in learning allows for achieving the same precision with less training data.

Nevertheless, the implementation of these algorithms requires substantial high-quality datasets and must ensure compliance with operational constraints in critical applications. Semi-supervised methods, as seen in [20], [24], mitigate this issue but miss out on leveraging graph networks' ability [25] to learn structured data representations, crucial for accuracy and efficiency in neural network-based solvers. Physically informed neural networks [24] emerge as a solution, reducing the dataset size needed for training and incorporating equations to guide network behavior.

While both PINNs and traditional machine learning methods can solve the OPF problem, they differ in approach. Traditional methods rely on historical data to establish input-output relationships, potentially overlooking underlying physical laws. In contrast, PINNs integrate physical principles into their neural network architecture, allowing for more accurate and generalizable solutions. However, PINNs' complexity in development and training surpasses that of traditional methods.

The utilization of typed graph networks, as presented in this work, not only enhances accuracy and efficiency but also adds interpretability to the solutions'an essential aspect for gaining

Manuscript received 3 August 2023; revised 8 January 2024 and 3 March 2024; accepted 19 April 2024. Date of publication 26 April 2024; date of current version 27 December 2024. This work was supported in part by the National Council of Science and Technology (CONACYT), Mexico, under Scholarship 710033, and in part by the Spanish National Plan for Scientific and Technical Research and Innovation, under Project PID2019-104711RB-I00. This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ. Paper no. TPWRS-01211-2023. *(Corresponding author: José Antonio Domínguez-Navarro.)*

The authors are with the Electrical engineering, University of Zaragoza, Zaragoza 50018, Spain (e-mail: tb.lopgar@proton.me; jadona@unizar.es).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2024.3394371.

Digital Object Identifier 10.1109/TPWRS.2024.3394371

operators' confidence in the practical application of these methods. The proposed solution in this article eliminates the need for training data, allows for simple modeling, and embeds power flow equations within the neural network.

II. Methodology

This section explains the layout of the proposed physics-informed typed graph neural network (PI-TGN) based OPF solver. It starts out explaining the elements of the loss function, as one of the biggest challenges in solving the OPF problem with machine learning lies in complying with the presented constraints. This way, the chosen loss function greatly influences the structure of the model, which is described subsequently.

A. Loss Function

As mentioned previously, the OPF problem can have different optimization objectives, but in any case it is constrained by physical laws governing the system and the operational constraints of the different elements that make up the electrical network. In this work, the objective is set to minimize the power losses in the transmission branches. To incorporate the different requirements of the OPF problem, the loss function used for training is multi-target and composed of three elements: one that depends on the objective, i.e. active power line losses, a second that aims at eliminating active and reactive power imbalance at each bus, and a final component that depends on the operation constraints of voltage magnitude and power generation of all buses. Each of the three terms has a corresponding weight, represented as λ_a, λ_b and λ_c. These weights are assigned to give more or less importance to each term in the total loss function.

The total active power loss in each sample is calculated as the sum of the difference between sent power and received power of all branches. These values are dependent on the inferred voltage magnitude and phase values, and on the branch characteristics, as shown below:

$$J_{\text{loss}} = \sum_{e=1}^{N_E} ||P_{f_e}\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right)| - |P_{t_e}\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right)|| \tag{1}$$

where N_E represents the number of branches of the transmission system; P_{f_e} and P_{t_e} represent the power flowing from the sending bus and to the receiving bus of each branch, respectively. These values are determined by the characteristics of each branch, and the voltage magnitude and phase values at their extremities (the "sending" and "receiving" buses). The branch characteristics are summarized in the single variable $\epsilon_e = (\rho_e, \delta_e, B_e, \tau_e, \omega_e)$; where ρ_e and δ_e represent the magnitude and phase of the series admittance of each branch e, B_e represents the total line charging susceptance, and in the case of transformers, τ_e and ω_e represent the nominal turns ratio and the phase shift angle. The voltage magnitude and phase values of the buses on each extremity of the branch are summarized in the variable $\Gamma_e = (V_{f_e}, \theta_{f_e}, V_{t_e}, \theta_{t_e})$; where V and θ represent the voltage magnitude and phase, respectively; the subindexes, fe and te, refer to whether it is on the sending or receiving side of the branch e. The voltage magnitude and phase are inferred values, and are thus marked with a hat ($\hat{\cdot}$), while the branch characteristics are given values, thus marked with a dot ($\dot{\cdot}$).

The total loss is calculated for every sample in the training batch, consisting of H samples. This way, to infer voltage magnitude and phase values that will minimize the branch losses, the first part of the loss function is simply the mean square deviation from zero loss of each sample, as presented below:

$$\mathcal{L}_a = \lambda_a \frac{1}{H} \sum_{h=1}^{H} J_{\text{loss,h}}^2 \tag{2}$$

The second part of the cost function is meant to enforce the equality constraint that enforces the fulfillment of the Kirchhoff laws, it is defined by the mean square deviation from the active and reactive power equilibrium point at each bus:

$$\mathcal{L}_b = \lambda_b \frac{1}{H} \sum_{h=1}^{H} \left(\frac{1}{N_B} \sum_{n=1}^{N_B} \left(\tilde{P}_{n,h}^2 + \tilde{Q}_{n,h}^2 \right) \right) \tag{3}$$

where N_B represents the number of substations in the electrical grid; $\dot{G}s_n$ and $\dot{B}s_n$ represent the bus shunt conductance and susceptance; $\tilde{P}_n$ and $\tilde{Q}_n$ represent the active and reactive power imbalances in the bus n, such that:

$$\begin{aligned} \tilde{P}_n = \; & \hat{P}g_n - \dot{P}d_n - \dot{G}s_n \hat{V}_n^2 \\ & - \sum_{e \in \mathcal{N}(n)} Pf_e\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right) - \sum_{e \in \mathcal{N}(n)} Pt_e\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right) \end{aligned} \tag{4}$$

$$\begin{aligned} \tilde{Q}_n = \; & \hat{Q}g_n - \dot{Q}d_n + \dot{B}s_n \hat{V}_n^2 \\ & - \sum_{e \in \mathcal{N}(n)} Qf_e\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right) - \sum_{e \in \mathcal{N}(n)} Qt_e\left(\hat{\Gamma}_e, \dot{\epsilon}_e\right) \end{aligned} \tag{5}$$

$$n = \; 1, \ldots, N_B$$

where the generated power ($\hat{P}g_n$ and $\hat{Q}g_n$) are inferred, the demanded power ($\dot{P}d_n$ and $\dot{Q}d_n$) are known, and the power from (Pf_e and Qf_e) and to (Pt_e and Qt_e) each bus of the branch e depends on the inferred voltage magnitude and phase values.

The final term of the loss function is in itself a composition of terms designed to discourage the transgression of the operational constraints. The violation degrees (VDs) are defined by ramp functions that depend on how much the inferred parameters deviate outside the minimum and maximum established limits, i.e. for any parameter $x \in \mathbb{R}$ with lower and upper limits $x^{\min}$, $x^{\max} \in \mathbb{R}$, the VDs for both limits are defined as:

$$\eta_{\min} = \begin{cases} 0, & \text{if } x^{\min} - x < 0 \\ x^{\min} - x, & \text{otherwise (limit violation)} \end{cases} \tag{6}$$

$$\eta_{\max} = \begin{cases} 0, & \text{if } x - x^{\max} < 0 \\ x - x^{\max}, & \text{otherwise (limit violation)} \end{cases} \tag{7}$$

These functions are applied to enforce the inequality constraints that are determined by the operational constraints of voltage magnitude, and power generation. For simplicity, the ramp function is expressed as the "max" function, which outputs the maximum of two given values.

$$\eta_{V\,\min} = \max(0, \dot{V}_n^{\min} - \hat{V}_n) \tag{8}$$

$$\eta_{V\,\max} = \max(0, \hat{V}_n - \dot{V}_n^{\max}) \tag{9}$$

$$\eta_{P\,\min} = \max(0, \dot{P}g_g^{\min} - \hat{P}g_g) \tag{10}$$

$$\eta_{P\,\max} = \max(0, \hat{P}g_g - \dot{P}g_g^{\max}) \tag{11}$$

$$\eta_{Q\,\min} = \max(0, \dot{Q}g_g^{\min} - \hat{Q}g_g) \tag{12}$$

$$\eta_{Q\,\max} = \max(0, \hat{Q}g_g - \dot{Q}g_g^{\max}) \tag{13}$$

$$n = 1, \ldots, N_B,\ g = 1, \ldots, N_G,$$

where N_B and N_G represent the number of buses and generators, respectively; (8) and (9) represent the VD of voltage magnitude from their lower and upper limits, respectively, for every bus; (10) and (11) represent the VD of the active power generation from their lower and upper limits, respectively, for every generation bus; (12) and (13) are similarly defined, but for the reactive power generation.

It should be noted that the ramp function is not differentiable in all its domain, but the gradient can be estimated through the subgradient. The two VDs of each parameter are aggregated, as shown below:

$$\eta_{V_{n,h}} = \eta_{V\,\min} + \eta_{V\,\max} \tag{14}$$

$$\eta_{P_{g,h}} = \eta_{P\,\min} + \eta_{P\,\max} \tag{15}$$

$$\eta_{Q_{g,h}} = \eta_{Q\,\min} + \eta_{Q\,\max} \tag{16}$$

As with the previous loss function expressions, the mean square deviation of each parameter is calculated as shown in (17)–(19), then all the resulting terms are added together in (20).

$$v_{\text{limit}} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_B}\sum_{n=1}^{N_B}(\eta_{V_{n,h}})^2 \tag{17}$$

$$p_{\text{limit}} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_G}\sum_{g=1}^{N_G}(\eta_{P_{g,h}})^2 \tag{18}$$

$$q_{\text{limit}} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_G}\sum_{g=1}^{N_G}(\eta_{Q_{g,h}})^2 \tag{19}$$

$$\mathcal{L}_c = \lambda_c(v_{\text{limit}} + p_{\text{limit}} + q_{\text{limit}}) \tag{20}$$

Equation (21) shows the final loss function, which is simply the sum of the terms described above ((2), (3) and (20)).

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_b + \mathcal{L}_c \tag{21}$$

B. PI-TGN Based OPF Solver

The proposed method for solving the OPF problem utilizes a PI-TGN neural network [26]. In [27], physics-guided neural networks emerge as a novel and competent alternative for optimization tasks in the real world. This approach models an electrical grid through a graph, incorporating various node types that mirror the grid's structure. It categorizes nodes into three types: two for buses (generator $\mathcal{V}_G$ and load $\mathcal{V}_D$) and one for electrical lines ($\mathcal{V}_E$), as depicted in Fig. 1. The topological connections among these different node types are indicated by two adjacency matrices, $\mathbb{A}_{g,e}$ and $\mathbb{A}_{d,e}$. Here, each node functions as a small neural network, and the exchange of

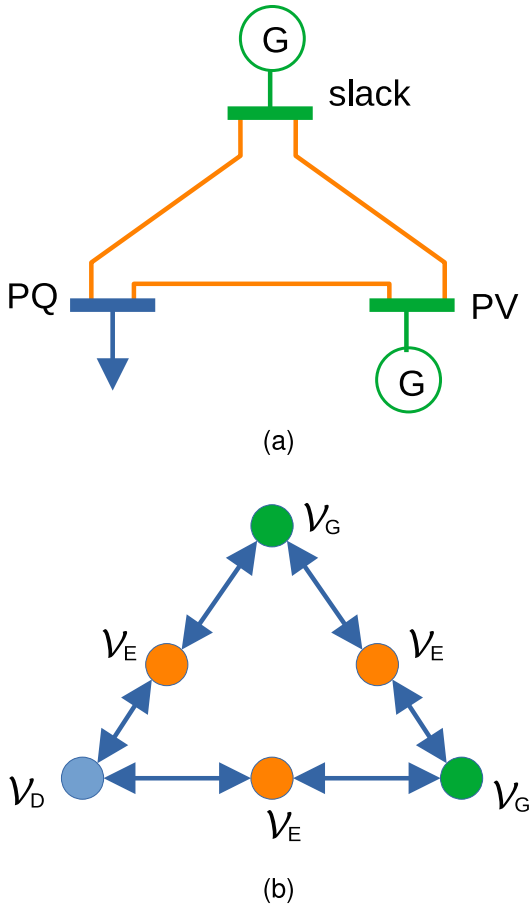

Fig. 1. (a) Small example of transmission grid and (b) corresponding PI-TGN architecture.

information between nodes is facilitated by messages leveraging the network's topological knowledge. Each layer of PI-TGN is defined by the graph $\mathcal{G} = (\mathcal{V}_D, \mathcal{V}_G, \mathcal{V}_E, \mathbb{A}_{g,e}, \mathbb{A}_{d,e})$.

The scheme of the PI-TGN based OPF solver proposed in this work is shown in Fig. 2. The solid blue squares represent the main parts of each PI-TGN layer, and other squares with a blue edge represent either components of the input or intermediate layer outputs (which are fed to the next layer). The orange square with the bright orange border shows the initialization of the inference variables, and is only input to the first PI-TGN layer. The orange square with purple edge represents the output only of the last PI-TGN layer. In the following subsections the different elements of the scheme are explained in more detail.

In Fig. 2, the visual representation of the neural network's inputs and outputs is provided, and a comprehensive summary of these elements is presented in Table I. The inputs consist of the grid state data, whose values are known, and the initial inference values of the control variables, whose values are unknown. The outputs comprise the values that are determined for these control variables.

C. Grid State Data and Initial Inference Values

The data used to train and verify the proposed model is obtained from a benchmark library from the IEEE Power & Energy Society task force on benchmarks for validation of emerging

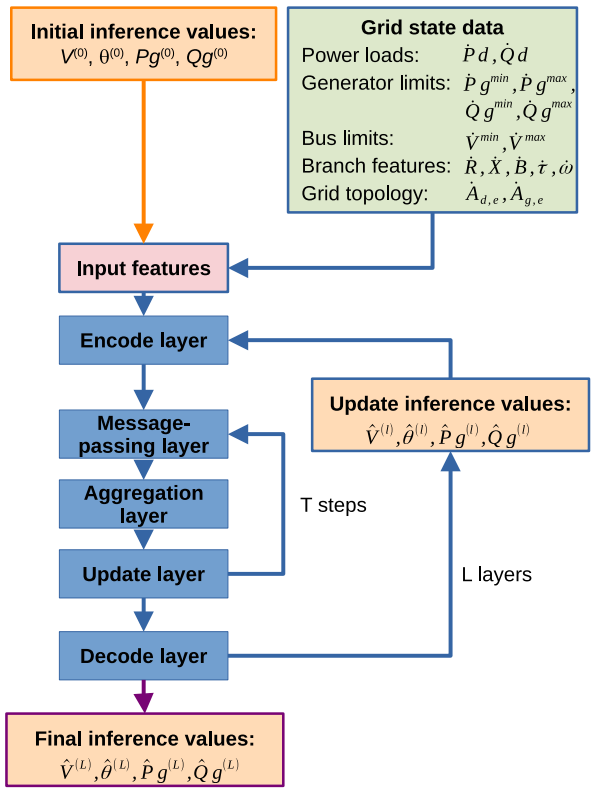

Fig. 2. PI-TGN based OPF solver scheme.

TABLE I
INPUT AND OUTPUT DATA

INPUTS	Nominal values	Limiting values
Bus data	Real and reactive power demand, shunt conductance and susceptance, voltage phase and magnitude.	Max. and min. voltage magnitude.
Generator data	Active and reactive power generation.	Max. and min. active and reactive generation.
Branch data	Sending/receiving buses, resistance, reactance, total line charging susceptance. **Transformers**: turns ratio, phase shift angle.	Not considered.
OUTPUTS	**Nominal values**	**Limiting values**
Bus data	Voltage phase and magnitude.	Not considered.
Generator data	Active and reactive power generation.	Not considered.

power system algorithms to solve the AC optimal power flow problem, posed as a non-convex, nonlinear program; the headers used were the ones of cases 14, 30, 57 and 200 buses [28]. All of the case files are curated in the Matpower data format, and thus include three main structures: bus data, generator data and transmission line data. The predefined, nominal and limiting values obtained are summarized in Table I.

To generate a batch of data, the first step is to establish the parameters that determine the grid state, either by adding uniform noise to specific elements of the benchmark data structures (bus, generator, branch), or by loading predefined time series data. The second step is to define initial voltage and power generation values. A batch of data is generated for every training iteration.

This way, to generate data for training uniform noise is added to the active and reactive power load of every bus, so that the final value lies between 80 and 120% of the original benchmark case value, uniform noise from 0.9 to 1.1 is also added to the shunt susceptance of every bus. To prepare the proposed model for possible changes in the power line parameters, which can be due to modifications in the operational conditions, added or removed vegetation and aging, branch noise is added to every training sample. Uniform noise is added to the resistance, reactance and line charging susceptance of every branch, resulting in values between 90 and 110% of the original benchmark values. Additionally, with the purpose of increasing the generalization capabilities of the proposed model to learn to deal with generators with different capacities, active power generation limits are varied between 90 and 110% of their original value. Once noise has been added to the injections (load characteristics) and to the transmission line characteristics, the topology of the grid is modified by randomly deleting a different transmission line for each sample.

The initial state of the parameters that will be inferred by the model are indicated by $Pg^{(0)}$, $Qg^{(0)}$, $V^{(0)}$ and $\theta^{(0)}$. The active and reactive power are independently initialized to a value between 25 and 75% of the previously established generation limits, i.e.

$$Pg_g^{(0)} = \mathrm{U}_{Pg} \cdot (\dot{P}g_g^{\max} - \dot{P}g_g^{\min}) + \dot{P}g_g^{\min} \tag{22}$$

$$Qg_g^{(0)} = \mathrm{U}_{Qg} \cdot (\dot{Q}g_g^{\max} - \dot{Q}g_g^{\min}) + \dot{Q}g_g^{\min} \tag{23}$$

$$g = 1, \ldots, N_G$$

$$\mathrm{U}_{Pg}, \mathrm{U}_{Qg} \sim \mathcal{U}(0.25, 0.75)$$

There must be some amount of balance between power generation and load in the initial state. In this work, a 5% error margin is set for this initial condition; if the total generation and load difference is larger, the initial generation state is scaled to meet the requirement.

The voltage magnitude of every bus is initialized to a value between 30 and 70% of the established bus limit, i.e.

$$V_n^{(0)} = \mathrm{U}_V \cdot (\dot{V}_n^{\max} - \dot{V}_n^{\min}) + \dot{V}_n^{\min} \tag{24}$$

$$n = 1, \ldots, N_B$$

$$\mathrm{U}_V \sim \mathcal{U}(0.3, 0.7)$$

The voltage phase of each bus is initialized to the slack bus angle established in the benchmark case, i.e.

$$\theta_n^{(0)} = \theta'_{\text{slack}}, \quad n = 1, \ldots, N_B \tag{25}$$

For the case in which established load time series are specified (e.g. for validation), these are simply loaded and used as they are, and random noise is added to the rest of parameters not given by the time series.

The process of modifying the benchmark electrical grid state by adding noise to the specified parameters and changing the topology of the grid, and initializing the voltage and power generation values is repeated every time a batch of data is needed by the proposed model, either for training, validation or testing.

D. Input Features

For the first PI-TGN layer, the input features are calculated from the grid state data (power loads, branch features and grid topology) and the initialization of the inference variables. For all the other PI-TGN layers, the input features depend on the grid state data and the updated inferred voltage and power generation values from the previous PI-TGN layer.

As was mentioned in Section II-A which discussed the loss function, the three main aspects of the OPF problem involve: the power losses in the transmission lines, the power equilibrium equations and the operational limits of the electrical grid. These components are correspondingly incorporated into the input features of the different types of nodes, such that the input feature set for every type of node is as shown below:

$$\mathbf{x}_g^{(l)} = \{\hat{V}_g^{(l-1)},\ \hat{\theta}_g^{(l-1)},\ \tilde{P}_g^{(l-1)},\ \tilde{Q}_g^{(l-1)},\ \eta_{V_g}^{(l-1)},\ \eta_{P_g}^{(l-1)},\ \eta_{Q_g}^{(l-1)}\}, \tag{26}$$

$$\mathbf{x}_d^{(l)} = \{\hat{V}_d^{(l-1)},\ \hat{\theta}_d^{(l-1)},\ \tilde{P}_d^{(l-1)},\ \tilde{Q}_d^{(l-1)},\ \eta_{Vd}^{(l-1)}\}, \tag{27}$$

$$\mathbf{x}_e^{(l)} = \{\epsilon_e,\ J_{\text{loss,e}}^{(l-1)}\}, \tag{28}$$

$$\forall g \in G,\ \forall d \in D,\ \forall e \in E$$

The inputs for generator type nodes include all their corresponding VDs ($\eta_{V_g}, \eta_{P_g}, \eta_{Q_g}$) since generator nodes must infer all the variables that are constrained by operation limits; the deviation from both active and reactive power equilibrium ($\tilde{P}_g,\ \tilde{Q}_g$) and the previously inferred voltage magnitude and phase ($\hat{V}_g,\ \hat{\theta}_g$) are also input features to this type of node. For load type nodes, only voltage values are inferred, and thus only the voltage magnitude VD (η_{V_d}) is considered as input, along with the deviation from active and reactive power equilibrium ($\tilde{P}_d,\ \tilde{Q}_d$) and the inferred voltage magnitude and phase ($\hat{V}_d,\ \hat{\theta}_d$). For branch type nodes, the branch power loss from the previous step ($J_{\text{loss,e}}$) and the branch characteristics (ϵ_e) which do not change from PI-TGN layer to layer, are used as input features.

E. PI-TGN Model

In this section, a description of the PI-TGN layer that is used sequentially and recursively a determined number of times, L, is described. The PI-TGN layer is composed of an encoder, a message-passing and aggregation step, an embedded update of all the nodes, and a decoder for the specified nodes with outputs. Regarding the graph representation, it should be noted that since in a transmission grid buses are not directly connected, but instead must be connected to at least one branch (islanding cases are not considered in this work), and each branch must be connected to two different buses, only two adjacency matrices are needed to represent the topology of the PI-TGN layer:

$$\mathbb{A}_{d,\,e} \in \mathbb{R}^{N_D \times N_E} \tag{29}$$

$$\mathbb{A}_{g,\,e} \in \mathbb{R}^{N_G \times N_E} \tag{30}$$

Where $\mathbb{A}_{d,\,e}$ represents the connectivity between load bus nodes and branch nodes, and similarly $\mathbb{A}_{g,\,e}$ represents the connectivity between generator bus nodes and branch nodes. The adjacency matrices are determined directly from the topology of the electrical grid to be modeled.

The process carried out in each PI-TGN layer begins with encoding the inputs of each type of node ($\mathbf{x}$); there are encoding MLPs defined for every type of node (γ_{ν_d}, γ_{ν_g} and γ_{ν_e}) that embed the input features into the latent space. This process is demonstrated in (31) for a PI-TGN layer l and a node type χ.

$$\nu_\chi^{(l)}(0) = \gamma_{\nu_\chi}(\mathbf{x}_\chi^{(l)}) \tag{31}$$

where $\nu_\chi^{(l)}(0) \in \mathbb{R}^{d_\chi}$ represents the initial embedded state of a χ type node.

After the initial encoding step, a predefined number T of message-passing and embedded update steps are carried out, as shown below:

$$\nu_d^{(l)}(t+1) = \sigma_{\nu_d}\left(\nu_d^{(l)}(t),\ \mathbb{A}_{d,\,e} \cdot \mu_{e\to d}\left(\nu_e^{(l)}(t)\right)\right) \tag{32}$$

$$\nu_g^{(l)}(t+1) = \sigma_{\nu_g}\left(\nu_g^{(l)}(t),\ \mathbb{A}_{g,\,e} \cdot \mu_{e\to g}\left(\nu_e^{(l)}(t)\right)\right) \tag{33}$$

$$\nu_e^{(l)}(t+1) = \sigma_{\nu_e}\left(\nu_e^{(l)}(t),\ \mathbb{A}_{d,\,e}^T \cdot \mu_{d\to e}\left(\nu_d^{(l)}(t)\right),\ \mathbb{A}_{g,e}^T \cdot \mu_{g\to e}\left(\nu_g^{(l)}(t)\right)\right), \tag{34}$$

$$\forall \nu_d \in \mathcal{V}_D,\ \forall \nu_g \in \mathcal{V}_G,\ \forall \nu_e \in \mathcal{V}_E$$

where $\nu_\chi^{(l)}(t) \in \mathbb{R}^{d_\chi}$ represents the t^{th} embedded state of a χ type node in PI-TGN layer l; σ_{ν_d}, σ_{ν_g} and σ_{ν_e} are the update MLPs for load, generation and branch nodes, respectively; $\mu_{e\to d}$ and $\mu_{e\to g}$ are the aggregation MLPs from branch type nodes to load and generator nodes, respectively; $\mu_{d\to e}$ and $\mu_{g\to e}$ are the aggregation MLPs from load and generator type nodes, respectively, to branch type nodes. These MLPs are the same for all layers.

Decoding functions are defined for load and generation type nodes (φ_{ν_d} and φ_{ν_g}), these functions serve to obtain the correct number of outputs. The generator type node decoding function has four outputs: voltage magnitude and phase changes, and active and reactive power generation changes. The load type node decoding function has two outputs: voltage magnitude and phase changes. The outputs of the decoder functions of a PI-TGN layer l for both types of nodes are shown below:

$$\mathbf{y}_d^{(l)} = \varphi_{\nu_d}\left(\nu_d^{(l)}(T)\right) = \{\Delta V_{\nu_d}^{(l)},\ \Delta\theta_{\nu_d}^{(l)}\} \tag{35}$$

$$\mathbf{y}_g^{(l)} = \varphi_{\nu_g}\left(\nu_g^{(l)}(T)\right) = \{\Delta V_{\nu_g}^{(l)},\ \Delta\theta_{\nu_g}^{(l)},\ \Delta Pg_{\nu_g}^{(l)},\ \Delta Qg_{\nu_g}^{(l)}\}$$
$$\forall \nu_d \in \mathcal{V}_D,\ \forall \nu_g \in \mathcal{V}_G \tag{36}$$

where ΔV_{ν_χ} and $\Delta\theta_{\nu_\chi}$ represent the voltage magnitude and phase modifications for a χ type bus; ΔPg_{ν_g} and ΔQg_{ν_g} represent the active and reactive power modifications for the generation type nodes.

F. Updated and Final Inference Values

As explained in the previous section, the decoder outputs of a certain PI-TGN layer are a modification value to be applied

to the inferred (or initial) state of the previous layer, for every bus node. The final inference values of a layer l are calculated as shown below:

$$\hat{V}_{\nu_d}^{(l)} = \hat{V}_{\nu_d}^{(l-1)} + \Delta V_{\nu_d}^{(l)} \tag{37}$$

$$\hat{\theta}_{\nu_d}^{(l)} = \hat{\theta}_{\nu_d}^{(l-1)} + \Delta \theta_{\nu_d}^{(l)} \tag{38}$$

$$\forall \nu_d \in \mathcal{V}_D$$

$$\hat{V}_{\nu_g}^{(l)} = \hat{V}_{\nu_g}^{(l-1)} + \Delta V_{\nu_g}^{(l)} \tag{39}$$

$$\hat{\theta}_{\nu_g}^{(l)} = \hat{\theta}_{\nu_g}^{(l-1)} + \Delta \theta_{\nu_g}^{(l)} \tag{40}$$

$$\forall \nu_g \in \mathcal{V}_G - slack$$

$$\hat{P}g_{\nu_g}^{(l)} = \hat{P}g_{\nu_g}^{(l-1)} + \Delta P g_{\nu_g}^{(l)} \tag{41}$$

$$\hat{Q}g_{\nu_g}^{(l)} = \hat{Q}g_{\nu_g}^{(l-1)} + \Delta Q g_{\nu_g}^{(l)} \tag{42}$$

$$\forall \nu_g \in \mathcal{V}_G$$

Thus, for a layer l, (37) and (39) represent the voltage magnitude output for load and generation nodes, respectively; (38) and (40) represent the voltage phase output for load and generation nodes, respectively; (41) and (42) represent the active and reactive power generation for generation nodes, respectively. Even though a slack node is not considered in this model, one of the generator buses is indeed the slack bus, as indicated by the benchmark case. For slack nodes, the output features corresponding to the voltage magnitude and phase are hard-coded to the upper limit for the magnitude, and to the original value of the slack for the phase, i.e.

$$\hat{\theta}_{\text{slack}}^{(l)} = \theta_{\text{slack}}^{(0)} \tag{43}$$

$$\hat{V}_{\text{slack}}^{(l)} = V_{\text{slack}}^{\max} \tag{44}$$

$$l = 1, \ldots, L$$

The complete PI-TGN layer-based model sequentially produces L outputs; the final outputs represent the final voltage and power generation inferences.

G. Training

The training process embodies unsupervised learning, where the network undergoes self-tuning of its weights. During this process, only input data is supplied, without corresponding output data. The algorithm autonomously adjusts its weights by minimizing the pre-defined loss function, ensuring adherence to the underlying physical principles.

The training process is described in Algorithm 1, where $\eta_{\min}^{(l)}$ and $\eta_{\max}^{(l)}$ summarize all VD values (voltage magnitude, active and reactive power generation). Similarly, $\mathbf{x}^{(l)} = \{\mathbf{x}_{\nu_g}^{(l)}, \mathbf{x}_{\nu_d}^{(l)}, \mathbf{x}_{\nu_e}^{(l)}\}$ represents the input features of all types of nodes of a layer l, and $\mathbf{y}^{(l)} = \{\mathbf{y}_{\nu_d}^{(l)}, \mathbf{y}_{\nu_g}^{(l)}\}$ represents the PI-TGN model outputs for load bus and generation bus type nodes. Lines 3 - 6 show the variables to which noise is applied, while line 7 clarifies which variables are left as the benchmark case. Line 8 indicates the initialization of the inference variables, and line 9 shows the moment in which the adjacency matrices can be calculated (which could be at any point after the configuration changes are applied). Lines 10–17 show the process that is repeated for each PI-TGN layer, specifically lines 11–13 represent the calculation of the loss function elements which are used to obtain part of the input features of the different types of nodes ((26)–(28)). The assembly of the input features is represented in line 14, and the outputs of the decoder function for the load and generation buses (after passing through the encoding, message-passing and updating iterations of each PI-TGN layer) are represented in line 15. Line 16 shows the updated inference variables of the PI-TGN layer, which will be used as inputs for the next layer. After the L PI-TGN layers are processed, the final loss function is derived and the gradients of the loss function with respect to the parameters of the NNs are calculated with the backpropagation algorithm. These gradients are used to modify the values of the NN parameters via the Adam optimization algorithm, afterwards a new batch of data is introduced to the PI-TGN model and the process is repeated until the maximum number of iterations is reached. The proposed PI-TGN OPF solver is trained in batches; and the encoding, message-passing, update and decoding NNs of the PI-TGN model are trained simultaneously.

Algorithm 1: Training Algorithm for PI-TGN Based OPF Solver.

1: REQUIRE: BUS, GENERATOR AND BRANCH DATA
2: WHILE $i < i_{\max}$
3: $\dot{P}d$, $\dot{Q}d$, $\dot{B}s \leftarrow$ Load noise
4: $\dot{P}g^{\min}$, $\dot{P}g^{\max} \leftarrow$ Active power limits noise
5: $\dot{R}$, $\dot{X}$, $\dot{B} \leftarrow$ Branch noise
6: $from$, $to \leftarrow$ Configuration change
7: $\dot{G}$, $\dot{V}^{\min}$, $\dot{V}^{\max}$, $\dot{Q}g^{\min}$, $\dot{Q}g^{\max}$, $\dot{\tau}$, $\dot{\omega} \leftarrow$ Benchmark
8: $\hat{V}^{(0)}$, $\hat{\theta}^{(0)}$, $\hat{P}g^{(0)}$, $\hat{Q}g^{(0)} \leftarrow$ Initialize
9: $\mathbb{A}_{d,e}$, $\mathbb{A}_{g,e} \leftarrow$ Adjacency matrices
10: FOR $l = 1 \ldots L$
11: $J_{\text{loss}}^{(l)} \leftarrow$ Branches power loss
12: $\tilde{P}^{(l)}$, $\tilde{Q}^{(l)} \leftarrow$ Power equilibrium
13: $\eta_{\min}^{(l)}$, $\eta_{\max}^{(l)} \leftarrow$ Violation degrees
14: $\mathbf{x}_{\nu_g}^{(l)}$, $\mathbf{x}_{\nu_d}^{(l)}$, $\mathbf{x}_{\nu_e}^{(l)} \leftarrow$ Input features
15: $\mathbf{y}^{(l)} \leftarrow TGN\_model(\mathbf{x}^{(l)}, \mathbb{A}_{d,e}, \mathbb{A}_{g,e})$
16: $\hat{V}^{(l)}$, $\hat{\theta}^{(l)}$, $\hat{P}g^{(l)}$, $\hat{Q}g^{(l)} \leftarrow$ Update inference variables
17: END FOR
18: $\mathcal{L}(J_{\text{loss}}^{(L)}, \tilde{P}^{(L)}, \tilde{Q}^{(L)}, \eta_{\min}^{(L)}, \eta_{\max}^{(L)}) \leftarrow$ Final loss value
19: $gradients \leftarrow \nabla_w(\mathcal{L})$
20: $weights \leftarrow w - \alpha \nabla_w(\mathcal{L})$
21: END WHILE
22: Save trainable weights

III. EXPERIMENTS

In this section, the predictive accuracy of the proposed PI-TGN-based OPF solver is evaluated by comparing it to the results obtained with the Matpower AC OPF solver. It also

TABLE II
TEST CASE CHARACTERISTICS

Test case	N_B	N_G	N_E	max ΔPd (MW)	max ΔQd (MVAr)	max ΔPg (MW)	max ΔQg (MVAr)
14_ieee	14	5	19	18.3	3.7	90.8	14.4
30_ieee	30	6	40	17.2	5.8	44.1	79.4
57_ieee	57	7	79	**74.7**	16.3	241.2	82.7
118_ieee	118	54	185	55.0	**22.4**	**301.3**	**450.3**
200_activ	200	49	244	14.4	4.11	205.3	65.3

TABLE III
OPF PREDICTION ERRORS (%)

	V	θ	Pg	Qg
14 buses	0.074	2.168	3.06	0.817
30 buses	0.056	1.554	1.98	3.03
57 buses	0.185	1.28	4.14	8.64
118 buses	0.079	1.06	13.31	14.58
200 buses	0.206	0.091	0.476	7.25

analyzes various design decisions in detail. All the instances of the proposed PI-TGN based OPF solver share the same hyperparameters:

- Number of PI-TGN layers: $L = 4$.
- Number of message passing and update steps: $T = 2$.
- Embedded dimension of all node types: $d = 16$.
- Learning rate: $\alpha = 1 \times 10^{-3}$.

A. Simulation Setup

To test the accuracy of the proposed solver, five different test cases are analyzed, each based on a header from the IEEE PES benchmarks [28]. The headers used are: "PGLIB OPF case14 IEEE", "PGLIB OPF case30 IEEE", "PGLIB OPF case57 IEEE", "PGLIB OPF case118 IEEE" and "PGLIB OPF case200 activ" [29]. The characteristics of each of the grids are given in Table II.

The noise added to each of the grids is specified in Section II-C, however, depending on the distribution and characteristics of loads and generators of each grid, the real load and generator noise applied to each of the grids differs. The last four columns of Table II show the maximum amount of load and generator alteration from the nominal case, in MW and $MVAr$, respectively. It can be seen, that even in the smallest grid of 14 buses, since three of the five generators are only for voltage support (they do not produce active power), all the active load has to be satisfied by only two generators, and so the noise applied to them results in large modifications from the nominal case. The opposite is true for the biggest grid of 200 buses, in which most of the generators produce active power and the load can be distributed between them, thus, the noise applied does not cause such big variations from the nominal case. As can be seen, in all cases the variation is significant, and is additional to the noise introduced in branch features, active power generation limits and shunt susceptance of the buses; the biggest differences from the nominal cases are shown in bold.

Furthermore, since in this work the branch flow limits are not considered, the branch ratings of the cases are manually modified to zero to make them unlimited when solving with the Matpower AC OPF solver, for comparison purposes. In the following tests, all Matpower results are included, even those in which some of the established limits are breached.

B. Prediction Accuracy With Noisy Benchmark Cases Test

For the first test, five instances of the proposed PI-TGN-based solver are trained on the five different electrical grid benchmark cases. The data is normalized using the per unit system, and a batch of 20 samples is taken for testing each instance.

Table III reports the average L2 loss (squared error loss) obtained from the difference between the inferred values obtained with the PI-TGN-based solver and the values obtained with the Matpower AC OPF solver using the IPOPT method. Each column shows the average predicted error in percentages: $\frac{1}{N_y}\sum(y-\hat{y})^2 \times 100$, for a variable y with N_y values.

For illustration purposes, Figs. 3 and 4 show the mean and standard deviation of the L2 loss of every bus of the batch for the test cases with 30 and 200 buses, respectively. The mean is represented with a blue line and the standard deviation with the green shaded area. Fig. 3(a), (b), (c), and (d) show the mean and standard deviation errors for the benchmark case with 30 buses, of voltage magnitude, phase, active power and reactive power, respectively. In this case all generators are capable of producing active power, and the generator most prone to divergence in the active and reactive power generated with respect to the Matpower case is generator 2. Fig. 4(a), (b), (c), and (d) show similar graphs, but for the benchmark case of 200 buses. In this case, most generators generate active power and are not too large; the largest generators are 29, 30 and 47, which is where the largest power generation errors are found. However, in general very good results were obtained with this case even though it is larger than the others. In all cases, the voltage phase of the slack bus is hard-coded to the nominal value, which is why the error in this point is close to zero. The voltage magnitude of the slack bus is set to the maximum limit, which in this case also results in a similar result to the Matpower solution.

In several trials, cases with many static synchronous or static var compensators in relation to active power generators seem to cause a negative effect on the predictive ability of the proposed model on the generator buses that have to compensate the load. The lack of active power generators and the large amount of noise from the nominal case is hypothesized to be the main reasons for this, although further studies should be carried out.

In all cases the best approximation is made of the voltage magnitude, and the power generation proves to be the most difficult to infer similarly to the Matpower solution.

C. Loss Function Elements With Noisy Benchmark Cases Test

Some attention must be paid to the influence of the different constraints included in the loss function to the results obtained with the PI-TGN-based solver. As there are no target values, the loss function elements greatly influence the behavior of the solver and may cause it to diverge from the Matpower results by giving more or less importance to different aspects.

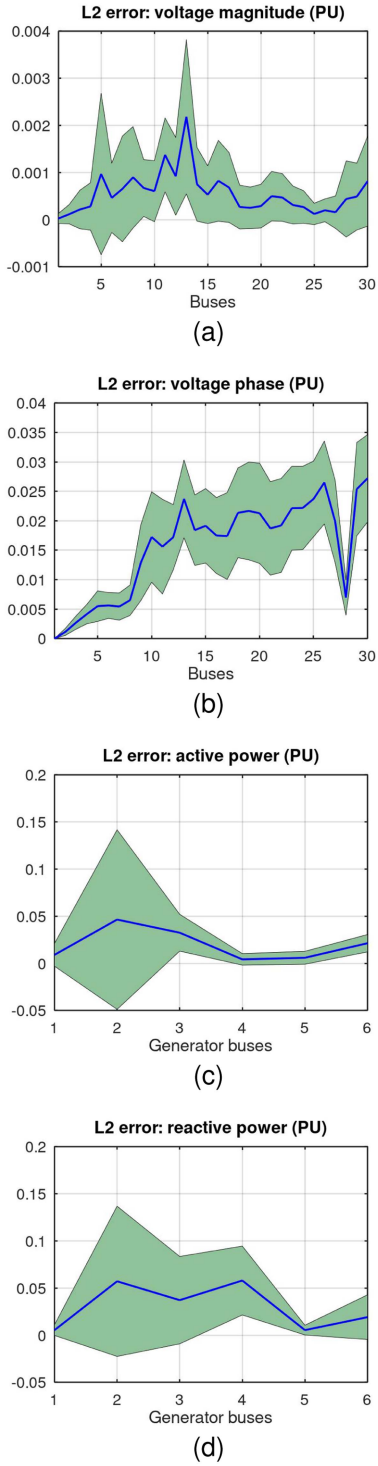

Fig. 3. Case 30 L2 error.

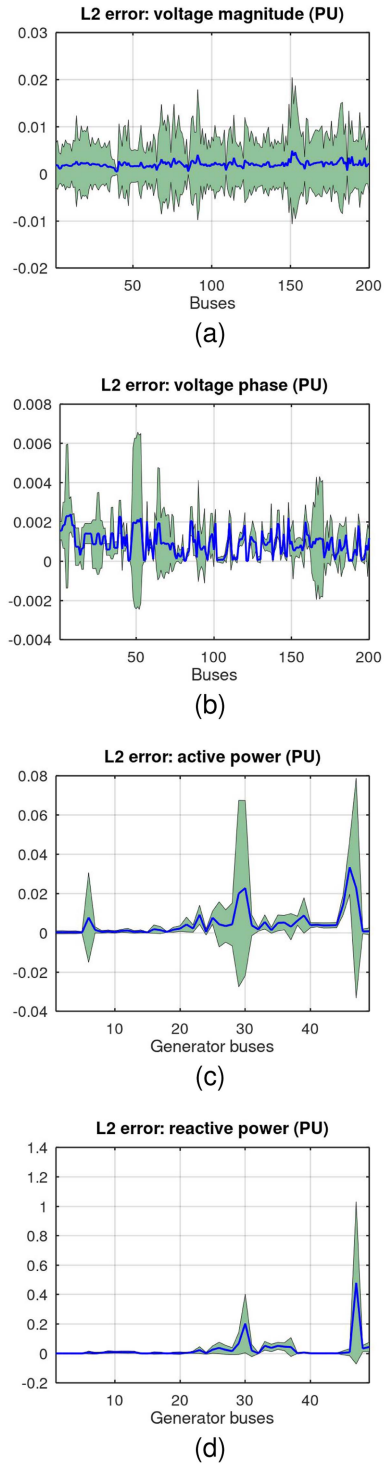

Fig. 4. Case 200 L2 error.

The main objective of the loss function is to minimize branch power losses. In Fig. 5 the total active power loss of all branches for every sample in the test batches is shown for both the PI-TGN-based solution and the Matpower solution. In this case, the loss is calculated as shown in (1); as can be seen, in general the total loss is lower with the proposed solution.

The violation of operational constraints is another element of the loss function. In the test case with the 30 bus grid, none of the operational limits are violated in the twenty test samples. Table IV shows the highest amount of operation constraint violation for the other test cases, for both power generation and voltage magnitude; where the "max" and "min" in the voltage magnitude column indicate whether it was an upper or lower limit that was violated.

It should be noted that, except for the 118 bus case, no more than five buses presented operational limit violations. In the 118

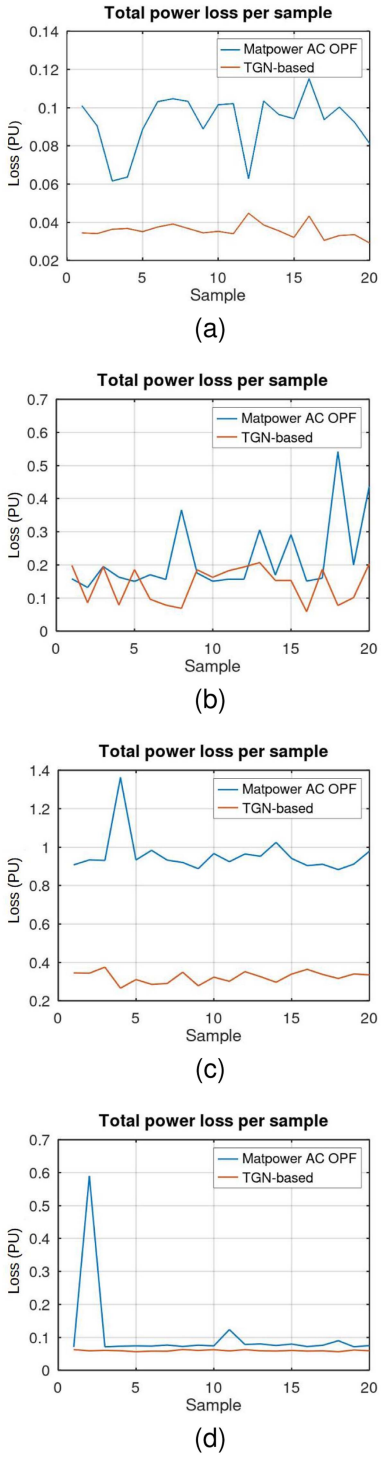

Fig. 5. Branch loss all cases.

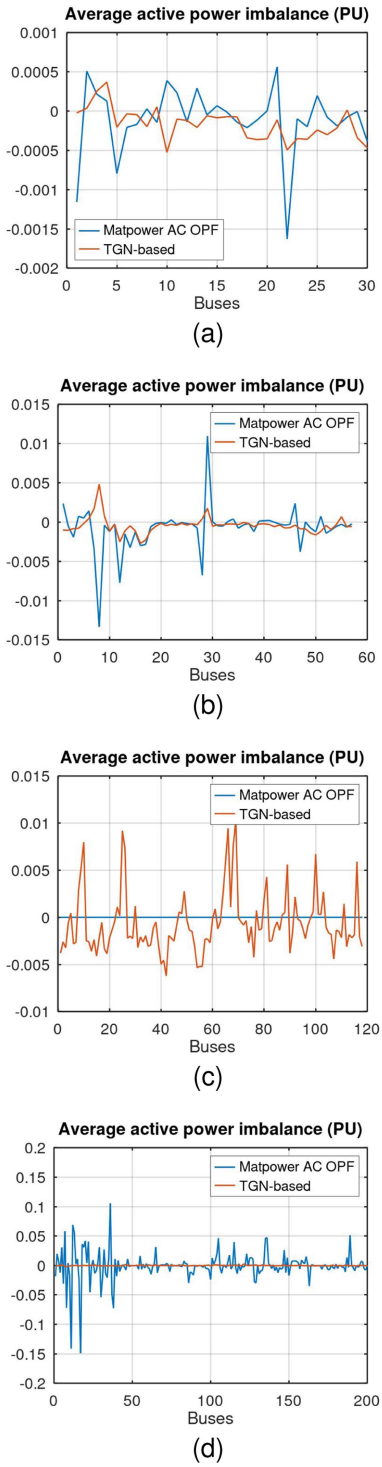

Fig. 6. Active power imbalance in all cases.

TABLE IV
OPERATIONAL LIMIT VIOLATIONS

	P (PU)	V max (PU)
57 bus case	≤ 0.0023	max: ≤ 0.0014
118 bus case	≤ 0.0038	max: ≤ 0.004
200 bus case	None	max: ≤ 0.01

bus case, around half of the buses exhibited a small amount of operational limit breach, this could also be caused by the high rate of compensators in the generator buses.

The last element of the loss function, which is essential, is ensuring the power balance at every bus. Figs. 6 and 7 show the average active and reactive power imbalance of every bus for all the test cases, for both the solutions from the proposed PI-TGN-solver and the solutions obtained with the Matpower

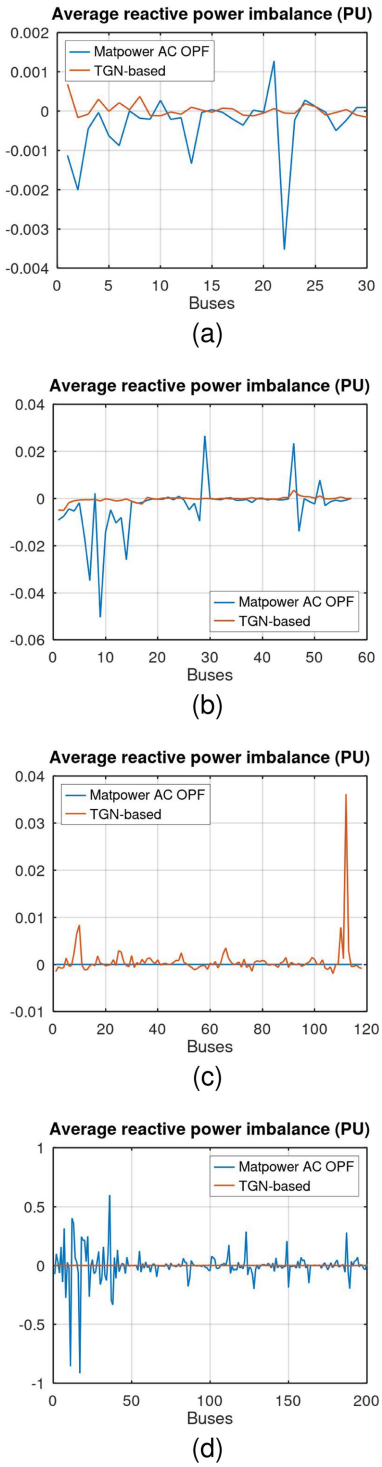

Fig. 7. Reactive power imbalance in all cases.

AC OPF solver. In most cases the imbalance of the proposed solver is smaller at every bus than the imbalance obtained with the Matpower solver, except for the case of 118 nodes, as can be seen in Figs. 6(c) and 7(c). On the other hand, for the 200 bus grid, the proposed solver successfully reduced the imbalance to virtually zero for all nodes, as observed in Figs. 6(d) and 7(d). The imbalances from the Matpower solutions could be due to the fact that all solutions were included, even those that did not converge in Matpower (no solution was found that could respect all the specified constraints), showing that the cases presented in the test were at times complicated to solve.

TABLE V
PREDICTION ERRORS IN TIME SERIES CASE (%)

Test case	V	θ	Pg	Qg
14_ieee (time series)	0.104	1.03	7.99	3.65

In general it can be observed that the proposed solver does a good job at minimizing the elements of the loss function, which emphasizes the importance of the physics-informed approach of adding the physical constraints into the loss function.

D. Prediction Accuracy With Time Series Case Test

In this subsection, instead of obtaining load data from benchmark cases, time series data from a 14 bus grid is used for training a distinct instance of the proposed PI-TGN-based OPF solver. The load profiles are obtained from the Codalab competition "Learning to Run a Power Network 2019" [30], which includes the active and reactive power load in intervals of 5 minutes. The only data considered in the time series is the active and reactive power load, the rest of the data is obtained from the benchmark case of 14 buses, adding noise just as with the noisy benchmark cases. Another important feature is that for every sample, a random branch is disconnected just as with the previous test cases, which would mean a very extreme case of a different branch outage every five minutes. The PI-TGN-based solver was trained and tested on batches of 300 samples (equivalent to one day and one hour of data). As with Table III, in Table V the average L2 loss (squared error loss) calculated from the difference between the results obtained with the proposed solver and the results obtained with Matpower using the interior point optimizer (IPOPT) method is reported. Each column shows the average predicted error of each of the inference variables, in percentages.

For this time series test the average and standard deviation of the L2 loss is also analyzed and shown in Fig. 8. Most results obtained are slightly better than with the ones obtained with noisy benchmark data for the 14 bus case, with the slack bus being the most prone to obtaining solutions that differ from the solution obtained with Matpower. Generator buses 3 to 5 do not produce active power which is why the active power error is zero for these elements.

E. Loss Function Elements of Test With Time Series Case

In this subsection the influence of the loss function elements on the time series case results are shown. The main objective function, which is to minimize the branch losses is represented in Fig. 9(a). It can be seen that the proposed solver underestimates power losses. Furthermore, the VDs are negligible (in the range of 1×10^{-5}) and are thus not included. The mean active and reactive power imbalance of every bus is shown in Fig. 9(b) and (c). As with the previous examples, the proposed solver achieves significantly lower active and reactive power imbalance in every

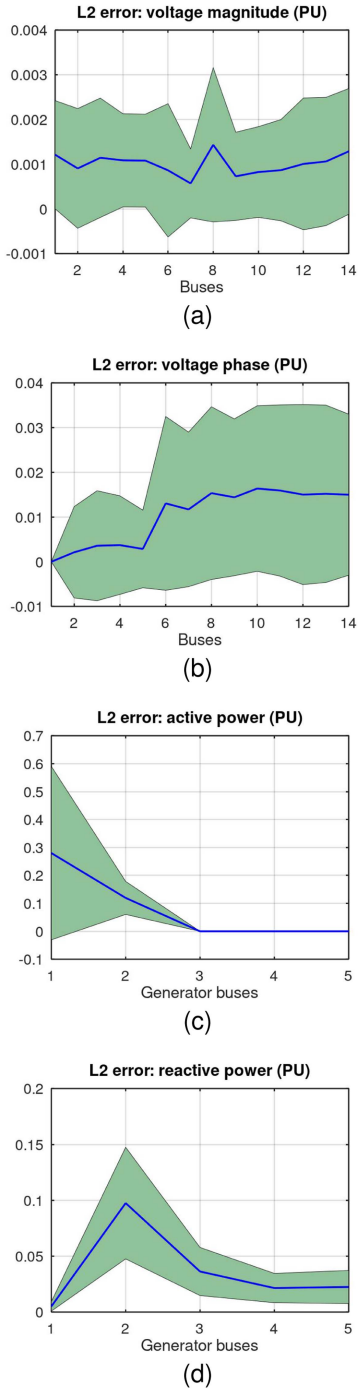

Fig. 8. Case 14 time series L2 error.

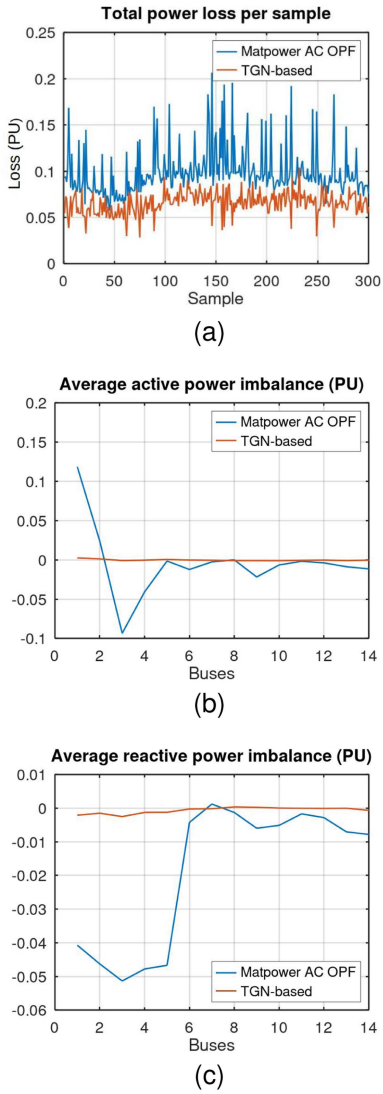

Fig. 9. Case 14 time series loss function elements.

TABLE VI
AVERAGE RUNTIME IN SECONDS

Test case	Matpower AC OPF solver	PI-TGN-based OPF solver
14_ieee	0.0328	1.77×10^{-3}
30_ieee	0.1059	1.798×10^{-3}
57_ieee	0.3327	2.389×10^{-3}
118_ieee	0.5263	3.297×10^{-3}
200_activ	4.6744	3.646×10^{-3}

node. It should be noted that in the time series case, with the added noise and network topologies, most of the samples did not converge in Matpower.

F. Time Considerations

Table VI illustrates the average time required to find an AC OPF solution with the Matpower AC OPF solver and the proposed PI-TGN-based OPF solver. It is important to note that because of the noise and configuration changes applied to the benchmark cases, the resulting cases are more challenging to solve than their original counterparts. As shown in Table VI, even for the smallest grid of only 14 nodes, the proposed solver is nearly 20 times faster than the conventional AC OPF solver. However, for larger grids the scalability of the proposed solution is emphasized. For the largest analyzed case of 200 buses, the proposed solver is more than 1000 times faster than the conventional solver. The computational time differences between classic methods and NN-based methods in the literature are

similar to those found in our calculations, often differing by one or two orders of magnitude. This aligns with data for classical algorithms [31] and neural network-based methods [32].

It should be noted that the time considered for the proposed solver only includes the computation time of the PI-TGN model with the different electrical grid sizes, but does not include the time it takes to add noise to the data or to generate the adjacency matrices.

Additionally, the training of all PI-TGN instances for this application was very fast, each instance took less than 4000 training iterations (less than 10 minutes) to minimize the loss function.

IV. Conclusion

In this chapter the potential of the proposed PI-TGN model applied to the AC-OPF problem is investigated. The AC-OPF problem involves finding the optimal values for variables that determine the power flow in an electric power system, while also taking into account the complex and changing interactions between voltage and power flow in the system and the physical and operational constraints that must be followed. This optimization problem is non-convex and nonlinear.

In this work, the goal of the established optimization problem is to minimize the amount of power lost as it is transmitted through the branches of the power system, while also taking into account the constraints related to the magnitude of the voltage and the operation of the power generation facilities. In order to account for the various constraints that must be considered when optimizing the power flow, the proposed PI-TGN model is trained with a loss function that incorporates several elements that not only considers the the main optimization objective, but also the established constraints.

The proposed model was tested on a number of different power networks of varying sizes, with a considerable amount of noise added to different variables (including loads, generators, and branch characteristics) and with changing electrical grid configurations in order to evaluate its performance. The tests focused on the ability of the proposed model to accurately predict the voltage magnitude and phase, and active and reactive power generation, while also considering the feasibility for use in real-world operations by analyzing the violation degrees of the operational constraints and the computational time. The model has proven to minimize the multi-target loss function, and to achieve results not very different from a conventional AC-OPF solver, while proving to be scalable and reducing considerably the computation time needed.

It is important to note that the proposed method is completely unsupervised, meaning that it does not rely on any target values in order to find a solution. While this makes the optimization process entirely self-contained, and allows it to be applied without the need of training data, the solutions are not always the same as the solutions obtained with reliable, conventional AC-OPF solvers. The proposed model could profit from including into the loss function an element that depends on target data in order to obtain results closer to those from an already trusted numerical method. Otherwise, more precise information from the physical system and the constraints could be added to the model inputs and to the loss function to improve the accuracy and to respect additional operational limits, as it was shown that the loss function elements were successfully minimized. In this work, the branch operational constraints were not considered, but could straightforwardly be added as additional violation degrees.

In this work, the different model instances were only tested on power networks of the same size as the ones they were trained on. The authors attempted to test the model on different size networks, but found that the performance was not satisfactory and that the differences between the networks were too great for the model to be able to generalize effectively. Additionally, improvement could be made in the implementation of the proposed method in order to test it on larger networks whose entire data sets may be too large to fit in the memory of a graphics processing unit (GPU).

The results of this study suggest that the proposed method may be a promising approach for approximating solutions to the AC-OPF problem, which is a fundamental component of many power system applications such as expansion planning and security assessments. These applications often require the use of a large number of simulations which can be time-consuming and resource-intensive. The solution to this problem will help to improve the efficiency of the power system operation and ensure that it is working within the established limits.

References

[1] J. Carpentier, "Contribution to the economic dispatch problem," *Bull. De la Societe Francoise des Electriciens*, vol. 3, pp. 431–447, 1962.

[2] W. Zhang and L. Tolbert, "Survey of reactive power planning methods," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, San Francisco, CA, USA, 2005, pp. 1430–1440. [Online]. Available: https://ieeexplore.ieee.org/document/1489402/

[3] W. Zhang, F. Li, and L. M. Tolbert, "Review of reactive power planning: Objectives, constraints, and algorithms," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2177–2186, Nov. 2007. [Online]. Available: https://ieeexplore.ieee.org/document/4349142/

[4] B. H. Chowdhury and S. Rahman, "A review of recent advances in economic dispatch," *IEEE Trans. Power Syst.*, vol. 5, pp. 1248–1259, Nov. 1990. [Online]. Available: https://ieeexplore.ieee.org/document/99376/

[5] M. Huneault and F. Galiana, "A survey of the optimal power flow literature," *IEEE Trans. Power Syst.*, vol. 6, no. 2, pp. 762–770, May 1991. [Online]. Available: https://ieeexplore.ieee.org/document/76723/

[6] J. A. Momoh, R. Adapa, and M. E. El-Hawary, "A review of selected optimal power flow literature to 1993. I. nonlinear and quadratic programming approaches," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–104, Feb. 1999. [Online]. Available: https://ieeexplore.ieee.org/document/744492/

[7] J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 105–111, Feb. 1999. [Online]. Available: https://ieeexplore.ieee.org/document/744495/

[8] K. Pandya, S. K. Joshi, K. S. Pandya, and S. K. Joshi, "A survey of optimal power flow methods," 2008. [Online]. Available: www.jatit.org

[9] X. Xia and A. Elaiw, "Optimal dynamic economic dispatch of generation: A review," *Electric Power Syst. Res.*, vol. 80, pp. 975–986, 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378779610000027

[10] F. Capitanescu, "Critical review of recent advances and further developments needed in ac optimal power flow," *Electric Power Syst. Res.*, vol. 136, pp. 57–68, 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378779616300141

[11] Z. Qiu, G. Deconinck, and R. Belmans, "A literature survey of optimal power flow problems in the electricity market context," in *Proc. IEEE/PES Power Syst. Conf. Expo.*, 2009, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/4840099/

[12] H. Abdi, S. D. Beigvand, and M. L. Scala, "A review of optimal power flow studies applied to smart grids and microgrids," *Renewable Sustain. Energy Rev.*, vol. 71, pp. 742–766, 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032116311583

[13] Z. Wang, A. Younesi, M. V. Liu, G. C. Guo, and C. L. Anderson, "AC optimal power flow in power systems with renewable energy integration: A. review of formulations and case studies," *IEEE Access*, vol. 11, pp. 102681–102712, 2023.

[14] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Trans.*, vol. 48, pp. 1172–1197, 2016. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/0740817X.2016.1189626

[15] M. Aien, M. Rashidinejad, and M. F. Firuz-Abad, "Probabilistic optimal power flow in correlated hybrid wind-pv power systems: A review and a new approach," *Renewable Sustain. Energy Rev.*, vol. 41, pp. 1437–1446, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032114007850

[16] F. Gao, Z. Xu, and L. Yin, "Bayesian deep neural networks for spatio-temporal probabilistic optimal power flow with multi-source renewable energy," *Appl. Energy*, vol. 353, 2024, Art. no. 122106.

[17] J. K. Skolfield and A. R. Escobedo, "Operations research in optimal power flow: A guide to recent and emerging methodologies and applications," *Eur. J. Oper. Res.*, vol. 300, pp. 387–404, 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221721008481

[18] A. Lotfi and M. Pirnia, "Constraint-guided deep neural network for solving optimal power flow," *Electric Power Syst. Res.*, vol. 211, 2022, Art. no. 108353.

[19] A. Kody, S. Chevalier, S. Chatzivasileiadis, and D. Molzahn, "Modeling the AC power flow equations with optimally compact neural networks: Application to unit commitment," *Electric Power Syst. Res.*, vol. 213, 2022, Art. no. 108282. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378779622004771

[20] F. Fioretto, T. W. 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. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5403

[21] Z. Yan and Y. Xu, "Real-time optimal power flow: A lagrangian based deep reinforcement learning approach," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3270–3273, Jul. 2020.

[22] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for DCOPF with generalization guarantees," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 719–730, Jun. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9599403/

[23] M. K. Singh, S. Gupta, V. Kekatos, G. Cavraro, and A. Bernstein, "Learning to optimize power distribution grids using sensitivity-informed deep neural networks," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9302942/

[24] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow," *Electric Power Syst. Res.*, vol. 212, 2022, Art. no. 108412.

[25] 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. [Online]. Available: https://ieeexplore.ieee.org/document/9053140/

[26] T. B. Lopez-Garcia and J. A. Domínguez-Navarro, "Power flow analysis via typed graph neural networks," *Eng. Appl. Artif. Intell.*, vol. 117, 2023, Art. no. 105567.

[27] J. Seo, "Solving real-world optimization tasks using physics-informed neural computing," *Sci. Rep.*, vol. 14, 2024, Art. no. 202.

[28] S. Babaeinejadsarookolaee et al., "The power grid library for benchmarking AC optimal power flow algorithms," 2019. [Online]. Available: http://arxiv.org/abs/1908.02788

[29] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, Jul. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7725528/

[30] A. Marot et al., "Learning to run a power network challenge for training topology controllers," *Electric Power Syst. Res.*, vol. 189, 2020, Art. no. 106635.

[31] M. Li, Y. Du, J. Mohammadi, C. Crozier, K. Baker, and S. Kar, "Numerical comparisons of linear power flow approximations: Optimality, feasibility, and computation time," in *Proc. IEEE Power & Energy Soc. Gen. Meeting*, 2022, pp. 1–5.

[32] H. Wu and Z. Xu, "Fast DC optimal power flow based on deep convolutional neural network," in *Proc. IEEE 5th Int. Elect. Energy Conf.*, 2022, pp. 2508–2512. [Online]. Available: https://ieeexplore.ieee.org/document/9846143/

Tania B. Lopez-Garcia received the M.Sc. degree in electrical engineering from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Mexico City, Mexico, in 2018, and the Ph.D. degree in renewable energies and energy efficiency from the University of Zaragoza, Zaragoza, Spain, in 2023, with a focus on transmission grid modeling and optimization via physics-informed graph neural networks. She has worked on analysis and optimization of renewable energy sources, specifically multi-criteria decision analysis of renewable energy sources and time series forecasting of loads. Her research interests include applications of data science and machine learning to power systems and local energy markets.

José Antonio Domínguez-Navarro (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Zaragoza, Zaragoza, Spain, in 2000. In 1992, he joined the University of Zaragoza, where he is currently a Full Professor with the Electrical Engineering Department. He has been involved in some research projects related to the optimization of power distribution networks, the modeling of electrical machines, and artificial intelligence applications in electrical engineering. His current research interests include the applications of soft computing in electric power systems, power distribution network planning, electric machines, and metaheuristic optimization.