Topology-aware Piecewise Linearization of the AC Power Flow through Generative Modeling

Young-ho Cho and Hao Zhu
Chandra Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX, USA
{jacobcho, haozhu}@utexas.edu

***Abstract*—Effective power flow modeling critically affects the ability to efficiently solve large-scale grid optimization problems, especially those with topology-related decision variables. In this work, we put forth a generative modeling approach to obtain a piecewise linear (PWL) approximation of AC power flow by training a simple neural network model from actual data samples. By using the ReLU activation, the NN models can produce a PWL mapping from the input voltage magnitudes and angles to the output power flow and injection. Our proposed generative PWL model uniquely accounts for the nonlinear and topology-related couplings of power flow models, and thus it can greatly improve the accuracy and consistency of output power variables. Most importantly, it enables to reformulate the nonlinear power flow and line status-related constraints into mixed-integer linear ones, such that one can efficiently solve grid topology optimization tasks like the AC optimal transmission switching (OTS) problem. Numerical tests using the IEEE 14- and 118-bus test systems have demonstrated the modeling accuracy of the proposed PWL approximation using a generative approach, as well as its ability in enabling competitive OTS solutions at very low computation order.**

***Index Terms*—Generative modeling, piecewise linear approximation, nonlinear AC power flow, grid topology optimization.**

I. Introduction

EFFECTIVE power flow modeling is critical for analyzing and optimizing large-scale power systems for efficient and reliable grid operations. With worldwide energy transitions and decarbonization, grid optimization tasks are increasingly challenged by e.g., uncertainty factors, extreme conditions, and fast computation needs. Meanwhile, optimizing the grid topology for increased flexibility has been advocated in problems like optimal transmission switching (OTS) [1], adaptive islanding [2], and post-disaster restoration [3]. Hence, it is important to develop effective power flow models that can facilitate the accurate and fast solutions for grid optimization problems, especially those with the combinatorial topology variables.

There exist significant efforts in developing approximation models for the nonlinear AC power flow. Notably, linearized power flow models have been advocated due to their simplicity, such as the well-known DC model [4], or the first-order approximation at an operating point for better accuracy [5]. As linear models are very limited by their generalizability across all possible operation regions [2], one straightforward extension is the piecewise linear (PWL) approximation approach by using multiple operating points; see e.g., [6]. By and large, there is a trade-off between accuracy and complexity for these model-based approaches, while the number and location of operating points could be difficult to select.

To tackle these issues, data-driven approaches have been recently advocated as an alternative for PWL modeling. Trained from realistic power flow scenarios, machine learning models such as K-plane regression [7] and neural networks (NNs) [8], [9] have shown good power flow approximation capabilities. In particular, ReLU-based NNs can be used to construct simple yet accurate PWL power flow models by incorporating the power flow Jacobian information [8]. Similar ideas have been explored in the constraint learning framework [9] but for general grid operational constraints. Interestingly, these PWL models under the ReLU activation allow to reformulate grid optimization problems with nonlinear power flow as mixed-integer linear programs (MILPs), for which there exist efficient off-the-shelf solvers. For example, successful applications to unit commitment and distribution management have been considered. Albeit the success, these existing approaches mainly build on an end-to-end learning framework that does not consider the underlying physical models of power flow. In addition, none of them has yet considered a flexible grid topology.

In this paper, we put forth a generative modeling approach to obtain the PWL approximation of AC power flow that is directly applicable to complex grid optimization problems. With the ReLU activation, we design the NN architecture to uniquely explore the generative structure of AC power flow. With the voltage and angle inputs, our proposed NN model first predicts the nonlinear terms that are common to all power variables in the first two layers, and then transforms these common terms to all line flows and power injections with two more layers. All the layers will be jointly trained to ensure an excellent consistency among all variables. This way, the proposed PWL model is generated in accordance with the power flow physics, while able to incorporate flexible topology connectivity. Thanks to our proposed NN design, we can cast grid optimization tasks like OTS into an MILP

This work has been supported by NSF Grants 1802319 and 2130706.

form for efficient solutions. We use the IEEE 14- and 118-bus test systems to validate the proposed PWL approximation in terms of improving power flow modeling accuracy over non-generative, as well as an excellent optimality/feasibility performance in solving AC-OTS.

The rest of the paper is organized as follows. Section II provides the nonlinear AC power flow modeling. In Section III, we develop the PWL model that uses a neural network and discuss the formulation steps to attain a mixed-integer program. Section IV provides the simulation set-up for the IEEE 14- and 118-bus test systems and presents the numerical comparisons and validations for the proposed scheme, along with some concluding remarks.

II. Nonlinear AC Power Flow Modeling

We first present the nonlinear AC power flow modeling for transmission systems, while introducing the relevant topology status variables and coupling terms useful for the discussions later on. Consider a transmission system consisting of n buses collected in the set $\mathcal{N} := \{1, \ldots, n\}$ and ℓ lines (including transformers) in $\mathcal{L} := \{(i,j)\} \subset \mathcal{N} \times \mathcal{N}$. For each bus $i \in \mathcal{N}$, let $V_i \angle \theta_i$ denote the complex nodal voltage phasor, and $\{P_i, Q_i\}$ denote the active and reactive power injections, respectively. For each line $(i,j) \in \mathcal{L}$, let $\theta_{ij} := \theta_i - \theta_j$ denote the angle difference between bus i and j, and $\{P_{ij}, Q_{ij}\}$ denote the active and reactive power flows from bus i to j; and similarly for $\{P_{ji}, Q_{ji}\}$ from bus j to i. In addition, the line's series and shunt admittance values are respectively denoted by $y_{ij} = g_{ij} + \mathrm{j} b_{ij}$ and $y_{ij}^{sh} = g_i^{sh} + \mathrm{j} b_i^{sh}$.

By defining a binary variable $\epsilon_{ij} \in \{0,1\}$ to indicate the status for each line $(i,j) \in \mathcal{L}$ (0/1: off/on), the nodal power balance at bus i per the Kirchhoff's law becomes

$$P_i = \textstyle\sum_{(i,j)\in\mathcal{L}} \ \epsilon_{ij} P_{ij}, \tag{1a}$$
$$Q_i = \textstyle\sum_{(i,j)\in\mathcal{L}} \ \epsilon_{ij} Q_{ij}. \tag{1b}$$

Note that these binary variables $\{\epsilon_{ij}\}$ will be important for formulating topology-related grid optimization tasks such as the optimal transmission switching problem as detailed later on. Without any topology changes, they can be fixed at $\epsilon_{ij} = 1$. For each line $(i,j) \in \mathcal{L}$, the power flows relate to the angle difference θ_{ij} and nodal voltages $\{V_i, V_j\}$, and in the case of transformer, its tap ratio a_{ij}, as given by

$$P_{ij} = V_i^2 (\frac{g_{ij}}{a_{ij}^2} + g_i^{sh}) - \frac{V_i V_j}{a_{ij}} (g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij}), \tag{2a}$$
$$Q_{ij} = -V_i^2 (\frac{b_{ij}}{a_{ij}^2} + b_i^{sh}) - \frac{V_i V_j}{a_{ij}} (g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij}). \tag{2b}$$

For the transmission lines, we can simply set $a_{ij} = 1$. As for a transformer, the tap ratio is typically set within the range of $[0.9, 1.1]$ and it only affects the primary-to-secondary direction. Thus, for the power flows in the secondary-to-primary direction, one can use $a_{ij} = 1$ in (2).

One advantage of our proposed piecewise linear (PWL) approximation is to leverage the underlying coupling among active and reactive power flows. To this end, let us denote the three nonlinear terms in (2) by

$$\gamma_i := V_i^2, \ \rho_{ij} := V_i V_j \cos\theta_{ij}, \ \text{and} \ \pi_{ij} := V_i V_j \sin\theta_{ij}.$$

To form the bi-directional power flows $\{P_{ij}, Q_{ij}, P_{ji}, Q_{ji}\}$ per line (i,j), we only need the nonlinear terms $\{\gamma_i, \gamma_j, \rho_{ij}, \pi_{ij}\}$, and the mapping between the two groups of variables is simply linear. To represent the resultant linear relation of (2) in a matrix-vector form, let us concatenate all the power flow variables in $\boldsymbol{z}^{pf} \in \mathbb{R}^{4\ell}$, and all the injection ones in $\boldsymbol{z}^{inj} \in \mathbb{R}^{2n}$. In addition, let $\boldsymbol{\gamma} \in \mathbb{R}^n$, $\boldsymbol{\rho} \in \mathbb{R}^\ell$, and $\boldsymbol{\pi} \in \mathbb{R}^\ell$ denote the respective vectors for the three groups of common terms. This way, we have

$$\boldsymbol{z}^{pf} = \boldsymbol{W}^\gamma \boldsymbol{\gamma} + \boldsymbol{W}^\rho \boldsymbol{\rho} + \boldsymbol{W}^\pi \boldsymbol{\pi}, \tag{3a}$$
$$\boldsymbol{z}^{inj} = \boldsymbol{W}^\psi \boldsymbol{z}^{pf} \tag{3b}$$

where the weight matrices $\{\boldsymbol{W}^\gamma, \boldsymbol{W}^\rho, \boldsymbol{W}^\pi, \boldsymbol{W}^\psi\}$ are of appropriate dimension given by the known line parameters and line status variables in (2) and (1), respectively. Clearly, the three groups of common terms are sufficient for fully generating all power flow and injection quantities, and our proposed generative modeling will work by predicting these terms as the first step.

A. Linear Approximation

We discuss the linear approximation for the common terms, which will be used by the proposed PWL models. Linear approximation is a basic approach to deal with power flow nonlinearity, thanks to its simplicity and reasonable accuracy within a small region of the operating point. We will consider the first-order approximation method to attain a linearized modeling, while there also exist other popular methods such as fixed-point method [10].

For the squared voltage term, it can be approximated by $\hat{\gamma}_i = 2V_i - 1, \forall i \in \mathcal{N}$, based on a flat-voltage value. Of course, this linearized model can be improved by using the exact operating point if different from the flat-voltage profile. As shown in [2], the former already attains a very high accuracy for power systems with well-regulated bus voltages within the p.u. range of [0.94, 1.06]. Thus, for simplicity, this linear representation of $\hat{\boldsymbol{\gamma}}$ will be adopted in this work.

Nonetheless, the other two terms $\boldsymbol{\rho}$ and $\boldsymbol{\pi}$ are more complicated to approximate than $\boldsymbol{\gamma}$ due to the presence of angle differences. To this end, we consider the first-order approximation for the former at the operating point. To simplify the notation, let us use $[\boldsymbol{\rho};\ \boldsymbol{\pi}] = \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{2\ell}$ to represent the nonlinear mapping from the input $\boldsymbol{x}$, which consists of the voltage magnitude $\boldsymbol{V} = \{V_i\}_{i\in\mathcal{N}} \in \mathbb{R}^n$ and the angle difference $\boldsymbol{\theta} = \{\theta_{ij}\}_{(i,j)\in\mathcal{L}} \in \mathbb{R}^\ell$. With a fixed operating point denoted by $\boldsymbol{x}_o$, the first-order approximation becomes

$$\begin{aligned}[\hat{\boldsymbol{\rho}}; \hat{\boldsymbol{\pi}}] &= \boldsymbol{f}(\boldsymbol{x}_o) + \boldsymbol{J}(\boldsymbol{x}_o)(\boldsymbol{x} - \boldsymbol{x}_o) \\ &= \boldsymbol{f}(\boldsymbol{x}_o) + \boldsymbol{J}(\boldsymbol{x}_o)\Delta\boldsymbol{x}\end{aligned} \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{x}_o)$ denotes the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ evaluated at $\boldsymbol{x}_o$, while we use $\Delta\boldsymbol{x} := \boldsymbol{x} - \boldsymbol{x}_o$ for simplicity. The ensuing

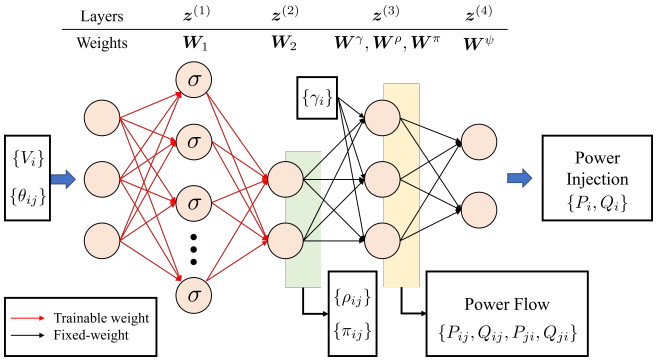


Fig. 1. The structure of the proposed neural network that can generate the common terms on the second layer through the trainable weight matrix and generate power flows and injections on the third and fourth layers through the fixed weight matrices.

section will build upon the linear model in (4) by using a data-driven approach to improve the approximation accuracy.

III. PWL Approximation via Generative Modeling

Our proposed PWL model uses a two-layer neural network (NN) to first approximate the common nonlinear terms in $[\boldsymbol{\rho}; \boldsymbol{\pi}]$, followed by two additional linear layers to generate the power flow and injection variables. As illustrated in Fig. 1, using the input voltage and angle difference in $\boldsymbol{x} = [\boldsymbol{V}; \boldsymbol{\theta}]$, the first two layers will rely on the ReLU activation functions to form the best PWL model for $[\hat{\boldsymbol{\rho}}; \hat{\boldsymbol{\pi}}]$ by adjusting the NN parameters. In addition, the last two layers use the fixed weight parameters from (3a) to generate the power flows for all lines, and accordingly, use (3b) to generate the power injection at all nodes. Note that the squared voltage $\hat{\boldsymbol{\gamma}}$ used by the third layer of generating power flow is based on the simple linearized model as described in Sec. II-A. Thus, the proposed approximation fully matches the power flow relations and coupling among different terms, in an efficient and generative fashion.

Using the ReLU activation, the first two layers can effectively produce a PWL mapping that can improve the accuracy of the linearized model in (4). The ReLU function is defined by $\sigma(\cdot)$ where outputs the entry-wise maximum between the input value and 0. Intuitively, when the activation status of ReLU function stays unchanged within a certain region of input values, its functional output enjoys the same linear relation with the input in that region. Therefore, one can view that the combination of ReLU activation status would divide the whole input space into multiple smaller regions, within each it boils down to a purely linear function. Thus, the overall function over the whole input space becomes a PWL one. In this sense, the number of linear regions will grow exponentially with the number of activation functions, and it would be challenging to search for all possible combinations. Therefore, we will train the NN parameters within the first two layers from generated data samples that can best select the activation status and linear regions from the data.

To concretely connect the NN model with PWL functions, we first consider a simple case of two linear regions obtained by using two different operating points, namely $\boldsymbol{x}_o$ and $\boldsymbol{x}_1$, as

$$[\hat{\boldsymbol{\rho}}; \hat{\boldsymbol{\pi}}] = \boldsymbol{f}(\boldsymbol{x}_o) + \Delta\boldsymbol{y}, \text{ with}$$
$$\Delta\boldsymbol{y} = \begin{cases} \boldsymbol{J}(\boldsymbol{x}_o)\Delta\boldsymbol{x}, & \boldsymbol{x} \in \mathcal{R}_o \\ \boldsymbol{J}(\boldsymbol{x}_1)(\boldsymbol{x} - \boldsymbol{x}_1) + \boldsymbol{r}, & \boldsymbol{x} \in \mathcal{R}_1 \end{cases} \tag{5}$$

where $\mathcal{R}_q$ represents the linear region corresponding to $\boldsymbol{x}_q$, and the residue in $\mathcal{R}_1$ is given by $\boldsymbol{r} := \boldsymbol{f}(\boldsymbol{x}_1) - \boldsymbol{f}(\boldsymbol{x}_o)$.

To recover the NN structure for (5), we follow [11] to assume that the two Jacobian matrices therein are different by a low-rank component. Specifically, we assume that

$$\boldsymbol{J}(\boldsymbol{x}_1) \approxeq \boldsymbol{J}(\boldsymbol{x}_o) + \boldsymbol{w}_2\boldsymbol{w}_1^\top \tag{6}$$

where both $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ consist of the NN weight parameters, that can be of much lower dimension than the size of Jacobian matrix. For simplicity, we consider both of them to be vectors with $\boldsymbol{w}_1 \in \mathbb{R}^{n+\ell}$ and $\boldsymbol{w}_2 \in \mathbb{R}^{2\ell}$, and thus the difference term in (6) becomes a rank-one matrix. This will be expanded to a higher-rank case later by using weight matrices. Interestingly, the simplification in (6) allows to unify the two scenarios in (5) by using one ReLU activation function, as given by

$$\Delta\boldsymbol{y} \approxeq \boldsymbol{J}(\boldsymbol{x}_o)\Delta\boldsymbol{x} + \boldsymbol{w}_2\sigma(\boldsymbol{w}_1^\top\Delta\boldsymbol{x} + b) \tag{7}$$

where b is a scalar bias parameter. When the ReLU function is not activated, it becomes the linear model in $\mathcal{R}_o$ of (5). Otherwise, upon the activation of $\sigma(\cdot)$ the resultant linear model should approach the one in $\mathcal{R}_1$ by recognizing the relation between the two Jacobian matrices in (6), as given by

$$\Delta\boldsymbol{y} \approxeq \left(\boldsymbol{J}(\boldsymbol{x}_o) + \boldsymbol{w}_2\boldsymbol{w}_1^\top\right)\Delta\boldsymbol{x} + \boldsymbol{w}_2 b. \tag{8}$$

In addition, to match the offset term in $\mathcal{R}_1$, we would need to have $\boldsymbol{w}_2 b \approxeq \boldsymbol{J}(\boldsymbol{x}_1)(\boldsymbol{x}_o - \boldsymbol{x}_1) + \boldsymbol{r}$. In general, the two-layer form in (7) may not fully express or match the two-region linearized model at the two operating points as in (5). Nonetheless, (7) definitely constitutes as a PWL approximation for the underlying $\boldsymbol{f}(\boldsymbol{x})$ function. In particular, the single ReLU activation in (7) has led to 2 linear regions for the resultant PWL model.

The simple case of two linear regions can be expanded to encompass more complex PWL model by increasing the number of ReLU activation functions. If the first layer has q ReLU functions with different linear transformations as the input, it is possible to generate a PWL model with up to 2^q linear regions. This way, the weight parameters form the two matrices $\boldsymbol{W}_1 \in \mathbb{R}^{(n+\ell)\times q}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{2\ell\times q}$, as well as the bias vector $\boldsymbol{b} \in \mathbb{R}^q$. The number of linear regions is related to the combination of activation status for all q ReLU functions. The larger q is, the more expressive the corresponding PWL model becomes, at the price of more model parameters to consider. This makes it difficult to determine the weight parameters using model-based linearization as in (5), motivating us to train these parameters from generated power flow samples.

In general, the latter can be designed to reflect the realistic operating points and the statistical variability around them, and thus the resultant PWL model could outperform a model-based approach by pre-selecting the points for linearization.

Before presenting the training loss, recall that the full generative model in Fig. 1 includes the first two layers for obtaining the nonlinear terms and two fixed-weight layers for power variables, as given by

$$\boldsymbol{z}^{(1)} = \sigma(\boldsymbol{W}_1^\top \Delta\boldsymbol{x} + \boldsymbol{b}), \tag{9a}$$

$$\boldsymbol{z}^{(2)} = \boldsymbol{f}(\boldsymbol{x}_o) + \Big(\boldsymbol{J}(\boldsymbol{x}_o)\Delta\boldsymbol{x} + \boldsymbol{W}_2\boldsymbol{z}^{(1)}\Big), \tag{9b}$$

$$\boldsymbol{z}^{(3)} = \boldsymbol{W}^\gamma\hat{\boldsymbol{\gamma}} + [\boldsymbol{W}^\rho; \boldsymbol{W}^\pi]\boldsymbol{z}^{(2)}, \tag{9c}$$

$$\boldsymbol{z}^{(4)} = \boldsymbol{W}^\psi\boldsymbol{z}^{(3)} \tag{9d}$$

where the first two layers generalize the simple case of (7) to q ReLU functions, and the last two layers follow from (3). When we generate random power flow data, the actual values for both $\boldsymbol{f}(\boldsymbol{x}) = [\boldsymbol{\rho}; \boldsymbol{\pi}]$ and $[\boldsymbol{z}^{pf}; \boldsymbol{z}^{inj}]$ can be obtained and using all of them for the loss function could effectively maintain the relations among the corresponding predicted values in (9). Specifically, we can use the Euclidean distance to form the following loss function

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{b}) =& \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^{(2)}\|_2^2 \\ &+ \lambda \Big\|[\boldsymbol{z}^{pf}; \boldsymbol{z}^{inj}] - [\boldsymbol{z}^{(3)}; \boldsymbol{z}^{(4)}]\Big\|_2^2\end{aligned} \tag{10}$$

where $\lambda > 0$ denotes a regularization hyperparameter to balance the error terms in different layers. The choice of λ can affect the prediction of the last two layers, and we will use a value larger than 1 to improve the accuracy in predicting $\boldsymbol{z}^{pf}$ and $\boldsymbol{z}^{inj}$. The average loss will be used to aggregate different samples, yielding the total training loss objective to minimize. After training, the proposed PWL model can fully generate the power flows and injections with linear transformations. On the other hand, (9a) is not a linear transformation due to the ReLU function. We also face nonlinear constraints when considering the binary line status variables with power flows. Hence, we will work to reformulate the ReLU function and the line status-related constraints into mixed-integer linear forms.

A. Mixed-integer Linear Formulation for the PWL Model

The proposed PWL models allow for formulating the nonlinear power flow equations into mixed-integer linear forms and thus enable efficient solutions for grid optimization problems involving topology variables. We present the formulation to attain a mixed-integer linear program (MILP) for the PWL modeling, and also for handling the line connectivity as in (1).

To adopt the PWL model in (9) into an MILP, the main issue lies in the ReLU function of (9a), as all other transformations are just linear ones. To tackle the ReLU function, we will use a technique based on the big-M tightening method [12]. For the k-th entry $z_k^{(1)}$ in (9a), we will approximate it by introducing a binary variable β_k, and its upper/lower bounds $\{\bar{M}_k, \underline{M}_k\}$. The two bounds can be determined through an off-line optimization procedure [13]. After determining these bounds and denoting the input in (9a) by $\hat{\boldsymbol{z}}^{(1)} = \boldsymbol{W}_1^\top\Delta\boldsymbol{x} + \boldsymbol{b}$, the big-M method asserts that each $z_k^{(1)}$ can be reformulated by using four linear inequality constraints, as given by

$$0 \le z_k^{(1)} \le \bar{M}_k\beta_k, \tag{11a}$$

$$\hat{z}_k^{(1)} \le z_k^{(1)} \le \hat{z}_k^{(1)} - \underline{M}_k(1-\beta_k), \tag{11b}$$

The binary variable β_k critically relates to the ReLU activation status based on the input $\hat{z}_k^{(1)}$. If the input $\hat{z}_k^{(1)} > 0$, then the constraints in (11b) enforce β_k to be one such that $z_k^{(1)} = \hat{z}_k^{(1)}$ holds exactly. Otherwise, if $\hat{z}_k^{(1)} \le 0$, the constraints in (11a) enforce β_k to be zero to yield $z_k^{(1)} = 0$. This way, the output $z_k^{(1)}$ from (11) exactly attains the ReLU-based output in (9a). Thus, with accurate upper/lower bounds, the big-M method allows for an equivalent reformulation of (9) into an MILP form.

Similarly, we formulate the line status-related constraints into an MILP form. We face the multiplication of continuous variables $\{P_{ij},\ Q_{ij}\}$ and binary variables ϵ_{ij} that are denoted as $\hat{P}_{ij} := \epsilon_{ij}P_{ij}$ and $\hat{Q}_{ij} := \epsilon_{ij}Q_{ij}$. To tackle this multiplication term, we will use the McCormick reformulation technique [14] derived from the big-M tightening method. After attaining the upper/lower bounds of active and reactive power flows $\{\bar{P}_{ij}, \underline{P}_{ij}\}$ and $\{\bar{Q}_{ij}, \underline{Q}_{ij}\}$, each $\{\hat{P}_{ij}, \hat{Q}_{ij}\}$ can be reformulated by using four linear inequality constraints, as given by

$$\underline{P}_{ij}\epsilon_{ij} \le \hat{P}_{ij} \le \bar{P}_{ij}\epsilon_{ij}, \tag{12a}$$

$$\underline{Q}_{ij}\epsilon_{ij} \le \hat{Q}_{ij} \le \bar{Q}_{ij}\epsilon_{ij}, \tag{12b}$$

$$P_{ij} + \bar{P}_{ij}(\epsilon_{ij}-1) \le \hat{P}_{ij} \le P_{ij} + \underline{P}_{ij}(\epsilon_{ij}-1), \tag{12c}$$

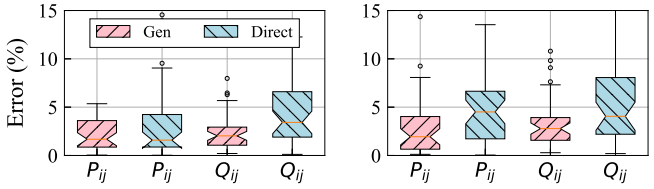
$$Q_{ij} + \bar{Q}_{ij}(\epsilon_{ij}-1) \le \hat{Q}_{ij} \le Q_{ij} + \underline{Q}_{ij}(\epsilon_{ij}-1). \tag{12d}$$

The outputs $\{\hat{P}_{ij}, \hat{Q}_{ij}\}$ critically relate to ϵ_{ij}. If ϵ_{ij} is equal to zero, then the constraints in (12a) and (12b) enforce $\hat{P}_{ij}$ and $\hat{Q}_{ij}$ to be zero. Otherwise, if ϵ_{ij} is equal to one, then the constraints in (12c) and (12d) enforce $\hat{P}_{ij}$ and $\hat{Q}_{ij}$ to be P_{ij} and Q_{ij}, respectively. This way, the output $\hat{P}_{ij}$ and $\hat{Q}_{ij}$ from (12) exactly attain the power flows based on the binary line status. Thus, the McCormick reformulation allows for an equivalent reformulation of the multiplication terms of power flows and line status variables into an MILP form.

IV. Numerical Studies

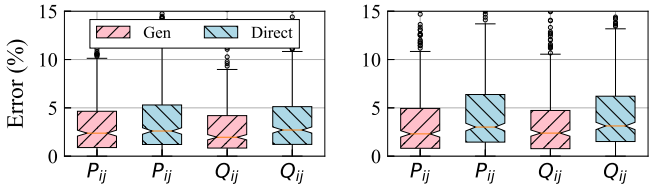
We have implemented the proposed generative modeling approach on the IEEE 14-bus and 118-bus test cases [15], to compare its performance in power flow modeling and grid topology optimization. The NN training has been performed in PyTorch with Adam optimizer on a regular laptop with Intel® CPU @ 2.70 GHz, 32 GB RAM, and NVIDIA® RTX 3070 Ti GPU @ 8GB VRAM. We have formulated the OTS problem through Pyomo [16] and used the Groubi optimization solver [17] for the resultant MILPs.

To train the proposed NN-based PWL models in Fig. 1, we generate 10,000 samples from the actual power flow model, with the outputs of common nonlinear terms $\{\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\pi}\}$, as



(a) Average Error (b) Maximum Error

Fig. 2. Comparisons of the (a) average error and (b) maximum error in approximating the line power flows for both the proposed Gen and Direct methods using the IEEE 14-bus system.



(a) Average Error (b) Maximum Error

Fig. 3. Comparisons of the (a) average error and (b) maximum error in approximating the line power flows for both the proposed Gen and Direct methods using the IEEE 118-bus system.

well as line flows and nodal injections. For each sample, we generate uniformly distributed voltage magnitudes within the range of [0.94, 1.06] p.u., and similarly for the angle, which randomly varies within $[-\pi/6, \pi/6]$ radians around the initial operating point. For the reference bus, namely, `Bus 1` in the 14-bus or `Bus 69` in the 118-bus system, we fix its voltage magnitude and angle at default values. For the first two trainable layers in Fig. 1, we use $q = 25$ and $q = 75$ ReLU activation functions, respectively for the two systems. The NN parameters have been trained via backpropagation using the loss function (10) with $\lambda = 10$, with a total of 20e3 epochs and a learning rate of 2.5e$-$3. Note that the λ value has been chosen through hyperparameter tuning. We separate 90% of the data set as training one and the rest 10% as testing one. The NN modeling results and error performance presented later on are based on the testing data only.

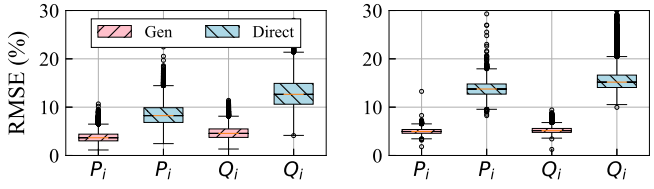
A. AC Power Flow Approximation

We first validate the AC power flow modeling performance of our PWL-based approximation. We compare the proposed generative modeling approach using the common nonlinear term prediction step (indicated by Gen) with the existing work [8] that directly predicts the power flow variables (indicated by Direct). The latter directly uses a two-layer NN of ReLU activation to output the line power flow, with the structure given by

$$\boldsymbol{z}^{pf} \leftarrow \boldsymbol{J}(\boldsymbol{x}_o)\Delta\boldsymbol{x} + \boldsymbol{W}_2\boldsymbol{z}^{(1)}. \tag{13}$$

Note that we use the same number of activation functions for both types of models.

We compare the approximation error between the predicted and actual line power flow values as normalized by the line capacity. Figs. 2 and 3 show the box plots of the normalized prediction error percentages of both active and reactive line



(a) IEEE 14 bus (b) IEEE 118 bus

Fig. 4. Comparisons of the root mean square error (RMSE) in predicting the injected power vectors for both the proposed Gen and Direct methods in the (a) IEEE 14-bus and (b) 118-bus systems.

flows, respectively for the 14- and 118-bus systems. Note that both the average error and the maximum error, out of all transmission lines in each system are included for comparisons. Each box plot shows the median values as midlines, the first and third quartiles as boxes, maximum values as horizon bars, and some outliers. Clearly, the proposed generative model is of better accuracy in predicting the line flows than the direct method, especially for the reactive power parts. Notably, the proposed method has shown significant improvements in terms of reducing the maximum values of errors in all cases. These results have verified the benefits of incorporating the underlying coupling between active and reactive power flows considered by our proposed NN design.

Furthermore, we also compare the error performance in predicting the active and reactive power injections, which can be formed directly from the line flows using (3b). Without any normalization basis, Fig. 4 instead shows the box plots for the root mean square error (RMSE) in predicting the injected power vectors in both test systems. Similar results have been observed for predicting the injections, with even more noticeable improvements in both active and reactive power values. This is because our proposed NN model in Fig. 1 has directly accounted for the power flow coupling, and thus its joint training process would achieve high consistency with nodal power balance. Thanks to the generative structure of the underlying NN design, our proposed PWL models can improve the accuracy and consistency in the resultant power flow approximation.

B. OTS Applications

We adopt the proposed PWL models in solving the OTS problem using the 118-bus system. The objective function and operational constraints are set up similarly to the typical optimal power flow (OPF) problem. Additionally, OTS allows for line switching under the constraint of a total switching budget of α lines, given by

$$\textstyle\sum_{(i,j)\in\mathcal{L}} \ \epsilon_{ij} \geq \ell - \alpha.$$

The number α is typically no greater than 5-10. Hence, the computation complexity of OTS is much higher than that of OPF, due to the integer line status variables. We introduce the proposed PWL models into the AC-OTS formulation by replacing the power flow constraints by (11), as well as the line status constraints by (12). This way, the resultant MILP problem can be efficiently solved with solvers like Gurobi.

We test the performance of the proposed PWL model-based OTS solutions. We compare it with the OTS solutions using the DC- and AC- power flow models, both provided by the open-source platform [18]. To compare across different OTS methods, we re-run the AC-OPF problem after fixing the topology with their line-switching decision outputs, using the MATPOWER [19] solver. This way, we can compare the metrics in terms of the objective costs (for optimality), as well as the percentage rates of infeasibility and constraint violations (for feasibility), using the corresponding AC-OPF outputs. For the two feasibility measures, the infeasible solution rates measure the percentage of infeasible solutions over all solutions, while the constraint violation rates are based on the percentage of over-limit voltage magnitude and angle over the infeasible solutions. Table I lists these metrics and also the computation time for each of the three OTS methods with a switching budget α equal to 1 or 3. Note that the computation time corresponds to solving the OTS optimization problem, not the follow-up AC-OPF one. A total of 1,000 power flow scenarios by having nodal demand uniformly distributed within [50%, 200%] of the initial demands [15] have been used to compute the average of all these four metrics. For the objective cost, the AC-OTS method has been used as a baseline (normalized to be 100%), and thus the other two OTS methods using approximate models attain higher percentage values. Nonetheless, the proposed PWL model only slightly increases the objective cost by less than 2%, while attaining exactly the same infeasiblity metrics as the AC-based OTS solutions. Notably, our model achieves a highly competitive performance and also great efficiency, as its computation time is almost a tenth of the AC-OTS one. In particular, the PWL model has allowed for a very low computation complexity in the order of DC-OTS one. But the latter leads to significantly worse feasibility performance, with an almost order of magnitude higher of infeasiblity rates than PWL-based OTS. Thanks to its high modeling accuracy, our proposed PWL model can greatly simplify the computation for grid topology optimization tasks by using the MILP reformulation trick, while approaching the ideal optimality/performance performance.

To sum up, we have designed a NN-based PWL approximation model for AC power flow with a good balance between model complexity and accuracy. Through its generative design, the proposed PWL models not only account for the underlying power flow coupling, but also allow for highly competitive solutions for topology-aware grid optimization problems. Our future research directions include improving the scalability of our proposed PWL models in large-scale power systems, as well as considering more generalized topology-aware grid optimization tasks like restoration and adaptive islanding.

TABLE I
COMPARISON OF THE OPTIMALITY AND FEASIBILITY OF THE SOLUTIONS OF THE AC, PWL, AND DCOTS.

	Switching Budget	AC	PWL	DC
Objective Cost (%)	$\alpha = 1$	100%	101.87%	102.67%
	$\alpha = 3$	100%	101.90%	104.21%
Infeasible Solution (%)	$\alpha = 1$	0.20%	0.20%	2.10%
	$\alpha = 3$	0.20%	0.20%	2.80%
Constraint Violation (%)	$\alpha = 1$	0.33%	0.33%	1.32%
	$\alpha = 3$	0.33%	0.33%	2.31%
Computation Time (s)	$\alpha = 1$	52.13 s	5.36 s	3.41 s
	$\alpha = 3$	57.67 s	6.43 s	3.89 s

REFERENCES

[1] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1346–1355, 2008.

[2] P. A. Trodden, W. A. Bukhsh, A. Grothey, and K. I. McKinnon, "Optimization-based islanding of power networks using piecewise linear AC power flow," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1212–1220, 2013.

[3] B. Chen, C. Chen, J. Wang, and K. L. Butler-Purry, "Sequential service restoration for unbalanced distribution systems and microgrids," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1507–1520, 2017.

[4] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[5] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.

[6] W. E. Brown and E. Moreno-Centeno, "Transmission-line switching for load shed prevention via an accelerated linear programming approximation of AC power flows," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2575–2585, 2020.

[7] J. Chen, W. Wu, and L. A. Roald, "Data-driven piecewise linearization for distribution three-phase stochastic power flow," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1035–1048, 2021.

[8] A. Kody, S. Chevalier, S. Chatzivasileiadis, and D. Molzahn, "Modeling the AC power flow equations with optimally compact neural networks: Application to unit commitment," in *2022 Power Systems Computation Conference (PSCC)*, 2022.

[9] G. Chen, H. Zhang, and Y. Song, "Efficient constraint learning for data-driven active distribution network operation," *IEEE Transactions on Power Systems*, 2023.

[10] J. W. Simpson-Porco, "A theory of solvability for lossless power flow equations—Part I: Fixed-point power flow," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1361–1372, 2017.

[11] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[12] I. Griva, S. G. Nash, and A. Sofer, *Linear and nonlinear optimization*. Siam, 2009, vol. 108.

[13] B. Grimstad and H. Andersson, "ReLU networks as surrogate models in mixed-integer linear programs," *Computers & Chemical Engineering*, vol. 131, p. 106580, 2019.

[14] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[15] [Online]. Available: http://labs.ece.uw.edu/pstca/

[16] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: modeling and solving mathematical programs in python," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 219–260, 2011.

[17] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[18] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "Powermodels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference (PSCC)*, 2018.

[19] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.