Modeling the AC Power Flow Equations with Optimally Compact Neural Networks: Application to Unit Commitment

Alyssa Kody†, Samuel Chevalier†, Spyros Chatzivasileiadis, Daniel Molzahn

***Abstract*—Nonlinear power flow constraints render a variety of power system optimization problems computationally intractable. Emerging research shows, however, that the nonlinear AC power flow equations can be successfully modeled using neural networks. These neural networks can be exactly transformed into mixed integer linear programs and embedded inside challenging optimization problems, thus replacing nonlinearities that are intractable for many applications with tractable piecewise linear approximations. Such approaches, though, suffer from an explosion of the number of binary variables needed to represent the neural network. Accordingly, this paper develops a technique for training an "optimally compact" neural network, i.e., one that can represent the power flow equations with a sufficiently high degree of accuracy while still maintaining a tractable number of binary variables. We demonstrate the use of this neural network as an approximator of the nonlinear power flow equations by embedding it in the AC unit commitment problem, transforming the problem from a mixed integer nonlinear program into a more manageable mixed integer linear program. We use the 14-, 57-, and 89-bus networks as test cases and compare the AC-feasibility of commitment decisions resulting from the neural network, DC, and linearized power flow approximations. Our results show that the neural network model outperforms both the DC and linearized power flow approximations when embedded in the unit commitment problem. The neural network formulation most often selects a feasible unit commitment schedule, and furthermore, it only selects an infeasible schedule if both the linear and DC methods are infeasible as well.**

***Index Terms*—AC power flow, AC unit commitment (AC-UC), mixed-integer linear program (MILP), neural networks, piecewise linear model**

I. Introduction

The AC power flow equations are routinely used to model network constraints in optimization problems related to the control, operation, and planning of power systems. These constraints, however, are both nonlinear and non-convex, resulting in optimization problems that can be NP-hard [1]. In practice, many power systems operation problems, like unit commitment (UC), optimal power flow (OPF), and optimal transmission switching (OTS), are solved using linearized approximations of the nonlinear AC power flow equations for the sake of computational tractability. However, linear approximations can result in suboptimal solutions or solutions that are infeasible in the original, nonlinear problem [2].

Piecewise linear models offer a method of improving solution accuracy by capturing some of the nonlinearity of the AC power flow equations [3], [4]. Using a conventional optimization solver to construct an *optimal* piecewise linear model of a large system of nonlinear power flow equations, however, is a challenging task: the associated program would need to simultaneously choose (i) where to perform explicit linearizations of the equations and (ii) how to define the hyperplanes which optimally separate the piecewise linear regions. Learning-based approaches offer a computationally tractable alternative. One way to construct such a piecewise linear approximation is through the use of neural networks (NNs) with rectified linear unit (ReLU) activation functions. Recent literature, e.g., [5], [6], has shown that NNs can be used to model the AC power flow equations with a high degree of accuracy. Furthermore, a NN using ReLU activation functions can be *exactly* transformed into a mixed integer linear program (MILP); this can be accomplished by representing the activation of each ReLU function as a binary variable and then using the big-M method to formulate activation constraints as a function of the NN weights and biases [7].

Although the exact MILP reformulation of a NN is often used for verification purposes [8], [9], it can also be embedded within a larger optimization problem as a function approximation. This technique allows us to transform a problem that may have originally been a challenging mixed integer nonlinear program (MINLP) into a more tractable MILP. Researchers have embedded NNs as MILPs within optimization problems for a variety of applications [10]–[13]. Specifically in the field of power systems, [14] and [15] encode frequency constraints using the MILP reformulation in microgrid scheduling and UC problems, respectively, and [8] encodes security constraints into the OPF problem.

In this paper, we develop a NN-based piecewise linear model of AC power flow equations that is both more accurate than a standard linearization and more computationally tractable than the original nonlinear equations. The resulting

† denotes an equal contribution among authors.

Alyssa Kody is an Argonne Scholar at the Argonne National Laboratory. Email: akody@anl.gov. A. Kody is supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DEAC02-06CH11357.

Samuel Chevalier and Spyros Chatzivasileiadis are with the Center for Electric Power and Energy, Department of Electrical Engineering, Technical University of Denmark (DTU). Emails: {spchatz; schev;}@elektro.dtu.dk. They are supported by the multiDC project as funded by Innovation Fund Denmark, Grant No. 6154-00020B, and by the ERC Project VeriPhIED, funded by the European Research Council, Grant Agreement No: 949899.

Daniel Molzahn is with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Email: molzahn@gatech.edu. D. Molzahn is supported by the NSF AI Institute for Advances in Optimization (#2112533).

NN can be embedded into any optimization problem as a surrogate power flow model via an exact MILP reformulation. However, the number of binary variables in the MILP reformulation *scales linearly* with the number of neurons. Therefore, embedding a NN containing many neurons, which may be necessary to represent high degrees of nonlinearity, can be a computational bottleneck. Grimstad and Andersson in [13] observe that "the feasibility of using the MILP formulation quickly fades with increasing network sizes."

This motivates the use of compression techniques to reduce the number of neurons in a NN, and consequently, the number of binary variables in the MILP reformulation. The machine learning community has developed a variety of methods to compress NNs with minimal accuracy compromise [16]–[18]. This paper specifically employs several methods in order to achieve an "optimally compact" NN model: (1) *low-rank updates*: we exploit the observation that, in most regions of practical interest, the power flow equations exhibit a relatively low degree of nonlinearity, and we therefore learn low-rank updates of a physics-based linearization of the AC power flow equations in these regions, (2) *pruning*: ReLU activation functions that are found to be always active or inactive over the input domain are fixed or removed, respectively, and (3) *sparsification*: we set NN weights below some threshold to be zero and re-train.

We demonstrate the use of the compact AC power flow NN by embedding it as a MILP within the day-ahead UC problem, which aims to determine the minimum cost generator commitment and dispatch while meeting load demand and abiding by physical laws and feasibility constraints. The AC unit commitment (AC-UC) problem, which constrains power injection and line flow quantities via the AC power flow equations, is a challenging nonconvex MINLP that is NP-hard [19]. In practice, utilities simplify the AC-UC problem to the DC unit commitment (DC-UC) problem, thus neglecting reactive power dispatch, line losses, and voltage magnitude constraints. Hence, corrective actions are often needed to account for these omissions [20]. Furthermore, as demonstrated in this paper, a commitment schedule based on a DC-UC solution can result in inoperable AC power dispatching solutions.

Researchers have proposed various methods to increase the computational tractability of the AC-UC problem using, for example, Lagrange relaxation [21], decomposition [20], [22], and convexification methods [23]. However, this is an ongoing research area without a singular superior solution technique identified yet. Recently, in [4], Nanou et al. develop a piecewise linear AC power flow model, which is embedded within a UC problem as a MILP, but their work does not use NNs or learning to generate the piecewise linear model. Although there are some works that focus on using NNs to solve the power flow equations [5], [6], [24], the authors are not aware of existing literature that embeds NN models of the AC power flow equations reformulated as MILPs within the UC problem, nor any other other power systems optimization problem. Accordingly, the contributions of this paper follow:

1) Using a sequence of feasible power flow solutions, we learn a piecewise linear power flow mapping based on low-rank updates of a physics-based linearization.
2) After transforming the learned model into an equivalent set of MILP constraints, we use iterative bound tightening, ReLU pruning, and parameter matrix sparsification in order to compress the effective size and complexity of the learned power flow model.
3) We pose a novel formulation of the AC-UC problem, where the AC power flow constraints are directly replaced by the piecewise linear power flow mapping.
4) Finally, we compare UC solutions using the NN-based piecewise linear power flow approximation to solutions using DC and linearized power flow approximations. Feasibility of the resulting commitment schedules are determined by solving a multi-time period AC-OPF (MTP AC-OPF) problem.

This paper is structured as follows. In Section II, we develop a NN-based piecewise linear power flow mapping based on low-rank linearization updates. We subsequently cast the learned model as a MILP and perform compression. In Section III, we pose standard UC formulations, and we show how our piecewise linear power flow mapping can be used to replace the standard power flow constraints. Then, in Section IV, we test the performance of the learned power flow mapping by comparing UC and MTP AC-OPF solutions collected from 14-, 57-, and 89-bus test cases. Finally, conclusions are offered in Section V.

II. Learning an Optimally Compact Power Flow Mapping

In this section, we first define a standard power flow model. Next, we develop a NN-based piecewise linear mapping of the power flow equations. This mapping is generated by a training procedure that learns optimal low-rank updates of a physics-based linearization. The resulting model is then reformulated as a MILP. Finally, we compress this model via bound tightening, ReLU pruning, and matrix sparsification.

A. Statement of Network Model

Consider a power network with bus set $\mathcal{N} = \{1, 2, ..., n\}$, line set $\mathcal{L} = \{1, 2, ..., m\}$, and signed incidence matrix $\boldsymbol{E} \in \mathbb{R}^{m \times n}$. Let $v, \theta, p^{\text{inj}}, q^{\text{inj}} \in \mathbb{R}^n$, which are the voltage magnitudes, voltage angles, real power injections, and reactive power injections, respectively. The nodal admittance matrix $\boldsymbol{Y}_b \in \mathbb{C}^{n \times n}$ relates complex nodal voltages and power injections via

$$p^{\text{inj}} + jq^{\text{inj}} = ve^{j\theta} \odot \left(\boldsymbol{Y}_b ve^{j\theta}\right)^*, \tag{1}$$

where Hadamard product $\odot$ performs component-wise multiplication. Complex line flows, in both directions, are related through line matrices $\boldsymbol{Y}_{\text{ft}}, \boldsymbol{Y}_{\text{tf}} \in \mathbb{C}^{m \times n}$:

$$p^{\text{ft}} + jq^{\text{ft}} = \boldsymbol{E}_{\text{ft}} ve^{j\theta} \odot \left(\boldsymbol{Y}_{\text{ft}} ve^{j\theta}\right)^* \tag{2}$$

$$p^{\text{tf}} + jq^{\text{tf}} = \boldsymbol{E}_{\text{tf}} ve^{j\theta} \odot \left(\boldsymbol{Y}_{\text{tf}} ve^{j\theta}\right)^*, \tag{3}$$

where $p^{\text{ft}}, q^{\text{ft}}, p^{\text{tf}}, q^{\text{tf}} \in \mathbb{R}^m$ are the active (p) and reactive (q) power line flows in the "from-to" (ft) and "to-from" (tf) directions. Furthermore, $\boldsymbol{E}_{\text{ft}} = \frac{1}{2}(|\boldsymbol{E}| + \boldsymbol{E})$, $\boldsymbol{E}_{\text{tf}} = \frac{1}{2}(|\boldsymbol{E}| - \boldsymbol{E})$

are matrices which select the sending end and receiving end voltages, respectively. Let $s^{\text{ft}}, s^{\text{tf}} \in \mathbb{R}^m$ be the apparent power flows in the "from-to" and "to-from" directions, respectively. Apparent power flows are related by $(s^{\text{ft}})^2 = (p^{\text{ft}})^2 + (q^{\text{ft}})^2$ and $(s^{\text{tf}})^2 = (p^{\text{tf}})^2 + (q^{\text{tf}})^2$.

B. A Low-Rank Piecewise Linear Power Flow Mapping Model

In order to mitigate the computational challenges associated with nonlinear power flow constraints, we use a NN to learn a piecewise linear power flow mapping. For notational convenience, we concatenate all of the active and reactive power injection and bidirectional apparent power flow equations into a function $f : \mathbb{R}^{\kappa=2n} \to \mathbb{R}^{\alpha=2n+2m}$, such that

$$f(v, \theta) \to (p^{\text{inj}}, q^{\text{inj}}, s^{\text{ft}}, s^{\text{tf}}). \tag{4}$$

By defining input vector $x = [v^T, \theta^T]^T$ and power flow output vector $y_{\text{pf}} = [(p^{\text{inj}})^T, (q^{\text{inj}})^T, (s^{\text{ft}})^T, (s^{\text{tf}})^T]^T$, a linearization of (4) yields $y_{\text{pf}} \approx f(x_0) + \boldsymbol{J}(x_0)\Delta x$. This may be rearranged to yield an affine transformation from x to y_{pf}:

$$y_{\text{pf}} \approx \boldsymbol{J}(x_0)x + \underbrace{f(x_0) - \boldsymbol{J}(x_0)x_0}_{r(x_0)}, \tag{5}$$

where $r(x_0)$ is a residual vector. In order to improve upon the predictive accuracy of the affine mapping in (5), an associated piecewise linear mapping from x to y_{pf} may be defined via

$$y_{\text{pf}} \approx \begin{cases} \boldsymbol{J}(x_0)x + r(x_0), & x \in \mathcal{R}_0 \\ \boldsymbol{J}(x_1)x + r(x_1), & x \in \mathcal{R}_1 \\ \quad \vdots & \\ \boldsymbol{J}(x_q)x + r(x_q), & x \in \mathcal{R}_q, \end{cases} \tag{6}$$

where $\mathcal{R}_0$, $\mathcal{R}_1$, ..., $\mathcal{R}_q$ represent the distinct regions of the linearization. Additionally, $\boldsymbol{J}(x_i)$ and $r(x_i)$, $x_i \in \mathcal{R}_i$ represent the Jacobian and constant terms, respectively, used to model the linear power flow mapping inside of region $\mathcal{R}_i$. Constructing a high-fidelity model of the form (6) is generally a challenging task, since selecting optimal points of linearization and separating hyperplanes is a nontrivial task.

Using a NN to directly model (6) can also be challenging, since transforming between Jacobian matrices in different regions requires a full-rank correction, i.e., $\boldsymbol{J}(x_1) = \boldsymbol{J}(x_0) + \boldsymbol{W}$, $\boldsymbol{W} \in \mathbb{R}^{\alpha \times \kappa}$, $\text{rank}\{\boldsymbol{W}\} = \min(\alpha, \kappa)$, and thus requires a potentially massive number of nonlinear activation functions. However, this correction can often be approximated by a *low-rank* surrogate, $\boldsymbol{W}_{\text{LR}}$, $\text{rank}\{\boldsymbol{W}_{\text{LR}}\} = \rho \ll \min(\alpha, \kappa)$. Low-rank matrices can always by decomposed into the outer product of two matrices, denoted here by $\boldsymbol{w}_1 \in \mathbb{R}^{\kappa \times \rho}$ and $\boldsymbol{w}_2 \in \mathbb{R}^{\alpha \times \rho}$, such that $\boldsymbol{W}_{\text{LR}} = \boldsymbol{w}_2 \boldsymbol{w}_1^T$.

In order to control how such low-rank updates are applied to a Jacobian, ReLU activation functions can be used. For example, consider the rank-1 ($\rho = 1$) update case; in this case, $\boldsymbol{w}_1$, $\boldsymbol{w}_1$ reduce to vectors w_1, w_2. If we define the hyperplane between two adjacent regions, e.g., $\mathcal{R}_0$ and $\mathcal{R}_1$, as $-b = w_1^T x$, then a piecewise linear prediction y_{pw} can be captured using a single ReLU activation function $\sigma(\cdot)$:

$$y_{\text{pw}} = \boldsymbol{J}_0 x + r_0 + w_2 \sigma(w_1^T x + b), \tag{7}$$

where $r_0 \triangleq r(x_0)$, $\boldsymbol{J}_0 \triangleq \boldsymbol{J}(x_0)$, etc. When the ReLU is not activated ($w_1^T x \leq -b$), the affine transformation of $\mathcal{R}_0$ in (6) is exactly recovered by (7). However, when the ReLU is activated ($w_1^T x > -b$), a rank-1 update is naturally applied to $\boldsymbol{J}_0$:

$$y_{\text{pw}} = \begin{cases} \boldsymbol{J}_0 x + r_0, & x \in \mathcal{R}_0 \\ \underbrace{(\boldsymbol{J}_0 + w_2 w_1^T)}_{\text{rank-1 update}} x + \underbrace{r_0 + w_2 b}_{\text{residual update}}, & x \in \mathcal{R}_1. \end{cases} \tag{8}$$

Generally, by choosing $\rho > 1$, higher-rank updates can be applied across more piecewise linear regions. To capture these updates, we approximate the full-order piecewise linear model stated in (6) via the function

$$y_{\text{pw}} = \boldsymbol{J}^\star x + r^\star + \boldsymbol{w}_2 \sigma(\boldsymbol{w}_1^T x + b), \tag{9}$$

where $\boldsymbol{J}^\star$ and $r^\star$ are the Jacobian and residual terms associated with a specified equilibrium point around which we learn low-rank updates. The model (9) may be interpreted as the application of low-rank updates to a full-rank, physics-based Jacobian. $\boldsymbol{J}^\star$ is the derivative of the physical power flow mapping in (4); therefore, it represents the concatenation of power injection and apparent power flow Jacobians: $\boldsymbol{J}^\star = [\boldsymbol{J}_{pq}^T \; \boldsymbol{J}_{s,\text{ft}}^T \; \boldsymbol{J}_{s,\text{tf}}^T]^T$; these are given in the Appendix.

The NN-based model of (9) consists of a physics-based affine feedthrough term ($\boldsymbol{J}^\star x + r^\star$) and a latent transformation term $\boldsymbol{w}_2 \sigma(\boldsymbol{w}_1^T x + b)$. While this latent term only has nonlinear activation functions applied to a single layer, the model can still provide a quantity of piecewise linearization regions which grows exponentially with the number of ReLUs.

Lemma 1. *If (9) contains ρ activation functions, then it can provide up to $q = 2^\rho$ distinct piecewise linearization regions.*

Proof. The activation of each ReLU generates a rank-1 update of $\boldsymbol{J}^\star$ and thus corresponds to a distinct piecewise linear region $\mathcal{R}_i$. The ρ independently controlled binary activation functions of (9) can therefore model 2^ρ piecewise linear regions. □

NNs that contain multiple layers of ReLU activation functions can also provide piecewise linear mappings. In this work, however, since we are targeting low-rank power flow approximations, we employ activation functions only on a single layer. For a given number of model parameters, shallow NNs cannot always achieve the same level of modeling power as a deeper NN. Results from [13], however, show that the MILP reformulation of a shallow NN solves faster than the MILP of a deeper NN (of equivalent complexity).

The model (9) is trained by first collecting input and output training data sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Next, an unconstrained optimization algorithm trains the NN by solving

$$\min_{b, \boldsymbol{w}_1, \boldsymbol{w}_2} \left\| \boldsymbol{Y} - (\boldsymbol{J}^\star \boldsymbol{X} + r^\star + \boldsymbol{w}_2 \sigma(\boldsymbol{w}_1^T \boldsymbol{X} + b)) \right\|_2^2. \tag{10}$$

C. Exact Neural Network Reformulation as MILP

Once trained, the NN-based model (9) can be reformulated as an equivalent set of MILP constraints [25]. Defining

intermediate variable $\hat{z} = \boldsymbol{w}_1^T x + b$, the ReLU function $z = \sigma(\hat{z}) \triangleq \max(\hat{z}, 0)$ is captured by the constraints

$$\begin{aligned} z_i &\leqslant \hat{z}_i - M_i^{\min}(1-\beta_i), \qquad & z_i &\geqslant \hat{z}_i \\ z_i &\leqslant M_i^{\max}\beta_i, & z_i &\geqslant 0, \end{aligned} \tag{11}$$

where $M_i^{\min}$ and $M_i^{\max}$ are the minimum and maximum values that $\hat{z}_k^i$ can take, respectively, and β is a vector of binaries: $\beta_i \in \{0,1\}$. The tightness of these big-M bounds influence the efficiency of the branch-and-bound algorithm used to handle these constraints [13]. With this formulation, the power flow mapping of (9) can be exactly captured via

$$y_{\text{pw}} = \boldsymbol{J}^\star x + r^\star + \boldsymbol{w}_2 z \tag{12a}$$
$$\hat{z} = \boldsymbol{w}_1^T x + b \tag{12b}$$
$$(11),\ \forall i \in \{1, \dots, \rho\}. \tag{12c}$$

Neglecting the NN, (12) reduces to a *linear* power flow model:

$$y_{\text{lin}} = \boldsymbol{J}^\star x + r^\star. \tag{13}$$

D. Compression of the NN-Based Power Flow Mapping

Once (9) has been reformulated as a MILP, it can be embedded as a constraint into a variety of optimization problems. To limit the computational complexity of the associated mixed integer constraints, however, we iteratively (i) sparsify the NN weighting matrices, (ii) tighten the big-M constraints associated with reformulation (11), and (iii) prune the NN's ReLUs. Each of these steps is summarized below.

1) NN Sparsification: Following the general procedure outlined in [25], we sparsify the NN weighting matrices $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ by setting some targeted percentage of the weights to 0. The weights selected are the ones which have the smallest absolute magnitude. After sparsification, the network is retrained; during retraining, sparsified entries are fixed to 0.

2) Big-M Bound Tightening: To tighten the big-M bounds, we first define inequality constraints associated with the NN inputs. That is, we define relevant nodal voltage and phase angle inequality constraints (e.g., $V_{\min} \leq v_i \leq V_{\max}$ and $|\theta_i - \theta_j| \leq \Delta\theta_{\max}$), denoted by $\mathcal{C}_v$ and $\mathcal{C}_\theta$. Next, we define inequality constraints associated with NN outputs, i.e., power injections and line flow limits, denoted by $\mathcal{C}_p$, $\mathcal{C}_q$, and $\mathcal{C}_s$. The lower bound $M_i^{\min}$ can be directly computed via the MILP

$$M_i^{\min} = \min_{x, y_{\text{pw}}, \beta} \ \hat{z}_i \tag{14a}$$
$$\text{s.t. } (12a) - (12c) \tag{14b}$$
$$x \in \{v, \theta \,|\, v, \theta \in \mathcal{C}_v, \mathcal{C}_\theta\} \tag{14c}$$
$$y_{\text{pw}} \in \{p^{\text{inj}}, q^{\text{inj}}, s_l \,|\, p^{\text{inj}}, q^{\text{inj}}, s_l \in \mathcal{C}_p, \mathcal{C}_q, \mathcal{C}_s\} \tag{14d}$$

where s_l denotes the apparent power line flows in both directions. The upper bound $M_i^{\max}$ may be computed by maximizing (14a), rather than minimizing it. We note that the constraint sets $\mathcal{C}_v, \mathcal{C}_\theta, \mathcal{C}_p, \mathcal{C}_q, \mathcal{C}_s$ can be naively defined using the engineering constraints associated with whatever optimization problem the NN is ultimately being used to solve (e.g., UC, OPF, etc.). Alternatively, these constraint sets can themselves be first tightened using, e.g., optimization-based bound tightening [26] or analytic methods [27].

3) ReLU Pruning: Using the calculated big-M bounds, individual ReLUs can be pruned (i.e., removed) from the NN. Pruning procedure: if $M_i^{\max} \leq 0$, then the associated ReLU is never active, and β_i is fixed to 0. However, if $M_i^{\min} > 0$, then the ReLU is always active, and β_i is fixed to 1.

III. Unit Commitment Formulations

In this section, we first present the three-binary formulation of the AC-UC problem, which is introduced in [28], using the indexing and notation schemes in [20]. We then present the UC problem where we approximate the AC power flows using the novel NN-based piecewise linear model, which has been exactly reformulated as the set of MILP constraints given in Section II-C. Next, we present two popular power flow approximations for comparison and benchmarking purposes. First, the AC-UC problem with linearized power flow equations and second, the DC-UC problem, where we neglect reactive power, line losses, and voltage magnitude deviations. Last, we present the MTP AC-OPF formulation. The UC schedules found using the three power flow approximation methods are tested for AC feasibility through checking for the feasibility of their corresponding MTP AC-OPF solutions.

A. Objective and Cost Constraints

Let $\mathcal{T} = \{1, \dots, T\}$ be the set of time indices, where each time index represents an hour of simulation and T is the total simulation time. Let $\mathcal{G} = \{1, \dots, G\}$ be the set of generators. For each generator $g \in \mathcal{G}$ and each time period $t \in \mathcal{T}$, there is a binary variable $y_{g,t} \in \{0,1\}$ indicating whether the generator is ON ($y_{g,t} = 1$) or OFF ($y_{g,t} = 0$). Let $y = \{y_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$ be the set of all generator statuses. Let $u_{g,t} \in \{0,1\}$ and $w_{g,t} \in \{0,1\}$ be the start-up and shut-down statuses, respectively, of generator $g \in \mathcal{G}$ at time $t \in \mathcal{T}$. If $u_{g,t} = 1$, then generator g starts-up at the beginning of hour t and is zero otherwise. If $w_{g,t} = 1$, then generator g shuts-down at the beginning of hour t and is zero otherwise. Let $u = \{u_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$ and $w = \{w_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$.

Let $c(\cdot)$ be the total cost of operating the network:

$$c(p^\Delta, u, w) = c^p(p^\Delta) + c^{su}(u, w), \tag{15}$$

where $c^p(\cdot)$ is the production cost and $c^{su}(\cdot)$ is the start-up cost. Let $p_{g,t}^\Delta \geqslant 0$ be the real power production of generator $g \in \mathcal{G}$ at time $t \in \mathcal{T}$ above $P_g^{\min} \geqslant 0$, the minimum real power production limit for generator g. Let $p^\Delta = \{p_{g,t}^\Delta \mid g \in \mathcal{G}, t \in \mathcal{T}\}$. The production cost $c^p(\cdot)$ is:

$$c^p(p^\Delta) = \sum\nolimits_{g \in \mathcal{G}} \sum\nolimits_{t \in \mathcal{T}} c_{g,t}^p(p_{g,t}^\Delta), \tag{16}$$

where $c_{g,t}^p(\cdot)$ is a convex piecewise linear function for all $g \in \mathcal{G}$ and $t \in \mathcal{T}$. The start-up cost $c^{su}(\cdot)$ has the form:

$$c^{su}(u, w) = \sum\nolimits_{g \in \mathcal{G}} \sum\nolimits_{t \in \mathcal{T}} c_{g,t}^{su}(u, w), \tag{17}$$

where $c_{g,t}^{su}(\cdot)$ is a monotonically increasing step function representing the startup costs for generator $g \in \mathcal{G}$, which increase with the amount of time the generator has been shut-down. See [20] for more details on this formulation.

B. Generation Constraints

For generator $g \in \mathcal{G}$, let the minimum uptime (the minimum period of time that the generator must be online before changing status) be T_g^u. The minimum downtime (the minimum period of time that the generator must be offline before changing status) is T_g^d. These constraints are given by:

$$\sum_{t'=t-T_g^u+1}^{t} u_{g,t'} \leqslant y_{g,t} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T} \tag{18}$$

$$\sum_{t'=t-T_g^d+1}^{t} w_{g,t'} \leqslant 1 - y_{g,t} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{19}$$

For values of $t' < 1$ (i.e., before the start of the simulation), we assume the values of $u_{g,t'}$ and $w_{g,t'}$ are known parameters.

Generators cannot start-up and shut-down in the same time period. This is enforced via (18) and (19) in conjunction with:

$$y_{g,t} - y_{g,t-1} = u_{g,t} - w_{g,t} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{20}$$

Let $r_{g,t} \geqslant 0$ be the real power reserve available to generator $g \in \mathcal{G}$ at time $t \in \mathcal{T}$, and let $r = \{r_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$. Parameter P_t^R is the total spinning reserve needed for the network at $t \in \mathcal{T}$, and we require that:

$$P_t^R \leqslant \sum\nolimits_{g \in \mathcal{G}} r_{g,t} \qquad t \in \mathcal{T}. \tag{21}$$

Let parameter $P_g^{\max} \geqslant 0$, be the maximum real power production limit for generator $g \in \mathcal{G}$. Parameters SU_g and SD_g are the maximum real power a generator $g \in \mathcal{G}$ can produce immediately after starting up and immediately before shutting down, respectively. Then, assuming $SU_g, SD_g \leqslant P_g^{\max}$, the start-up generation limits are enforced via the following constraints:

$$p_{g,t}^{\Delta} + r_{g,t} \leqslant (P_g^{\max} - P_g^{\min}) y_{g,t} - (P_g^{\max} - SU_g) u_{g,t} - (P_g^{\max} - SD_g) w_{g,t+1}, \forall g \in \{i \in \mathcal{G} \mid T_i^u \geqslant 2\}, t \in \mathcal{T} \tag{22}$$

$$p_{g,t}^{\Delta} + r_{g,t} \leqslant (P_g^{\max} - P_g^{\min}) y_{g,t} - (P_g^{\max} - SU_g) u_{g,t} \qquad \forall g \in \{i \in \mathcal{G} \mid T_i^u = 1\}, t \in \mathcal{T} \tag{23}$$

$$p_{g,t}^{\Delta} \leqslant (P_g^{\max} - P_g^{\min}) y_{g,t} - (P_g^{\max} - SD_g) w_{g,t+1} \qquad \forall g \in \{i \in \mathcal{G} \mid T_i^u = 1\}, t \in \mathcal{T}. \tag{24}$$

We also require a constraint to ensure the shut-down generation limits are enforced during the first time period $t = 1$:

$$w_{g,1} \leqslant 0 \qquad \forall g \in \{\, i \in \mathcal{G} \mid P_i^{\text{init}} > SD_i\}, \tag{25}$$

where P_i^{init} is the real power produced by generator $g \in \mathcal{G}$ the time period before the simulation begins. Parameters RU_g and RD_g are the real power ramp-up and ramp-down limits, respectively, for generator $g \in \mathcal{G}$. Generators must abide by ramping limits, which restrict the change in real power:

$$p_{g,t}^{\Delta} + r_{g,t} - p_{g,t-1}^{\Delta} \leqslant RU_g \qquad \forall g \in \mathcal{G}, t \in \mathcal{T} \tag{26}$$

$$-p_{g,t}^{\Delta} + p_{g,t-1}^{\Delta} \leqslant RD_g \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{27}$$

Generator $g \in \mathcal{G}$ has lower and upper reactive power limits, $Q_g^{\min}$ and $Q_g^{\max}$, respectively. Let $q_{g,t}$ be the reactive power production of generator $g \in \mathcal{G}$ at time $t \in \mathcal{T}$. The reactive power output of each generator is constrained to be within its limits when active:

$$Q_g^{\min} y_{g,t} \leqslant q_{g,t} \leqslant Q_g^{\max} y_{g,t} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{28}$$

C. AC-OPF Constraints

Recall that $\mathcal{N}$ is the set of buses in the network. Now, let $\mathcal{G}_b$ be the set of generators at bus $b \in \mathcal{N}$. Parameters $P_{b,t}^D$ and $Q_{b,t}^D$ are the real and reactive power demand at $b \in \mathcal{N}$ and $t \in \mathcal{T}$. Let $p_{b,t}^{\text{inj}}$ and $q_{b,t}^{\text{inj}}$ be the real and reactive power, respectively, *injected into the network* from bus $b \in \mathcal{N}$ at time $t \in \mathcal{T}$. Then, the real power balance constraints are:

$$p_{b,t}^{\text{inj}} = \sum_{g \in \mathcal{G}_b} \left(p_{g,t}^{\Delta} + P_g^{\min} y_{g,t}\right) - P_{b,t}^D \quad \forall b \in \mathcal{N}, t \in \mathcal{T}, \tag{29}$$

and the reactive power balance constraints are:

$$q_{b,t}^{\text{inj}} = \sum_{g \in \mathcal{G}_b} q_{g,t}^G + \sum_{g \in \mathcal{SC}_b} q_{g,t}^{SC} - Q_{b,t}^D \quad \forall b \in \mathcal{N}, t \in \mathcal{T}. \tag{30}$$

Recall that $\mathcal{L}$ is the set of transmission lines in the network. Each line $\ell \in \mathcal{L}$ has a designated "from" bus and "to" bus, which can be arbitrarily chosen. $s_{\ell,t}^{\text{ft}}$ is the apparent power flow on line $\ell \in \mathcal{L}$ at time $t \in \mathcal{T}$ with flow from the "from" bus and to the "to" bus; $s_{\ell,t}^{\text{tf}}$ is defined oppositely. The apparent power flows on each line $\ell \in \mathcal{L}$ cannot exceed their maximum allowable limit $S_l^{\max}$:

$$s_{\ell,t}^{\text{ft}} \leqslant S_l^{\max}, \quad s_{\ell,t}^{\text{tf}} \leqslant S_l^{\max} \qquad \forall l \in \mathcal{L}, t \in \mathcal{T}. \tag{31}$$

Let $v_{b,t}$ be the voltage magnitude at bus $b \in \mathcal{N}$ and time $t \in \mathcal{T}$. Parameters $V_b^{\max}$ and $V_b^{\min}$ are the maximum and minimum voltage magnitude limits, respectively, for bus $b \in \mathcal{N}$. Then, the constraints on the voltage magnitudes are as follows:

$$V_b^{\min} \leqslant v_{b,t} \leqslant V_b^{\max} \qquad \forall b \in \mathcal{N}, t \in \mathcal{T}. \tag{32}$$

We designate the "from" and "to" bus voltage angles of line $\ell \in \mathcal{L}$ as $\theta_{\ell_{\text{f}}}$ and $\theta_{\ell_{\text{t}}}$, respectively. Parameters Θ_ℓ^{min} and Θ_ℓ^{max} are the minimum and maximum voltage angle differences for line $\ell \in \mathcal{L}$. We constrain the voltage angle differences as:

$$\Theta_\ell^{min} \leqslant \theta_{\ell_{\text{f}}} - \theta_{\ell_{\text{t}}} \leqslant \Theta_\ell^{max} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T}. \tag{33}$$

Last, we set the voltage angle of the reference bus to zero for all $t \in \mathcal{T}$. Let $b_{\text{ref}} \in \mathcal{N}$ designate the reference bus. Then:

$$\theta_{b_{\text{ref}},t} = 0 \qquad \forall t \in \mathcal{T}. \tag{34}$$

D. AC-UC Formulation

Let $p_{\ell,t}^{\text{ft}}$, $p_{\ell,t}^{\text{tf}}$, $q_{\ell,t}^{\text{ft}}$, and $q_{\ell,t}^{\text{tf}}$ be the the real and reactive power flows on lines $\ell \in \mathcal{L}$ at time $t \in \mathcal{T}$. Now, let $\mathcal{L}_b^{\text{ft}} \subseteq \mathcal{L}$ and $\mathcal{L}_b^{\text{tf}} \subseteq \mathcal{L}$ be the subsets of lines that are originating and terminating, respectively, at bus $b \in \mathcal{N}$. Parameters G_b^{sh} and B_b^{sh} are the shunt conductance and susceptance, respectively, at bus $b \in \mathcal{N}$. Then, the power injected into the network from bus $b \in \mathcal{N}$ is equivalent to:

$$p_{b,t}^{\text{inj}} = G_b^{\text{sh}} v_{b,t}^2 + \sum\nolimits_{\ell \in \mathcal{L}_b^{\text{ft}}} p_{\ell,t}^{\text{ft}} + \sum\nolimits_{\ell \in \mathcal{L}_b^{\text{tf}}} p_{\ell,t}^{\text{tf}} \tag{35}$$

$$q_{b,t}^{\text{inj}} = -B_b^{\text{sh}} v_{b,t}^2 + \sum\nolimits_{\ell \in \mathcal{L}_b^{\text{ft}}} q_{\ell,t}^{\text{ft}} + \sum\nolimits_{\ell \in \mathcal{L}_b^{\text{tf}}} q_{\ell,t}^{\text{tf}}. \tag{36}$$

The apparent power flows are equal to:

$$s_{\ell,t}^{\text{ft}} = \sqrt{(p_{\ell,t}^{\text{ft}})^2 + (q_{\ell,t}^{\text{ft}})^2} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T} \tag{37}$$

$$s_{\ell,t}^{\text{tf}} = \sqrt{(p_{\ell,t}^{\text{tf}})^2 + (q_{\ell,t}^{\text{tf}})^2} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T}. \tag{38}$$

Then, the MINLP formulation of the AC-UC problem is:

$$\min_{\mathcal{X}^{\text{cont}},\mathcal{X}^{\text{bin}},\theta,v,p^{\text{ft}},p^{\text{tf}},q^{\text{ft}},q^{\text{tf}}} (15) \text{ s.t. } (18)-(38), \qquad \text{(AC-UC)}$$

where we collect all the continuous generation variables in the set $\mathcal{X}^{\text{cont}} = \{r_{g,t}, p^{\Delta}_{g,t}, q_{g,t}, q^{SC}_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$, and all the binary commitment decision variables in the set $\mathcal{X}^{\text{bin}} = \{y_{g,t}, u_{g,t}, w_{g,t} \mid g \in \mathcal{G}, t \in \mathcal{T}\}$. Note that each variable corresponding to the individual elements of vectors p^{ft}, p^{tf}, q^{ft}, q^{tf}, v, and θ are optimization variables as well.

E. UC using Power Flow Approximations

We model power flow using the piecewise linear NN model described in Section II-B. The corresponding MILP model is presented in (12). Then, the NN-based AC-UC problem is:

$$\min_{\mathcal{X}^{\text{cont}},\mathcal{X}^{\text{bin}},\theta,v} (15) \text{ s.t. } (12), (18)-(33). \qquad \text{(NN AC-UC)}$$

A linearized power flow model is presented in (13). Then, the following MILP is the version of the AC-UC problem using this linearized power flow model:

$$\min_{\mathcal{X}^{\text{cont}},\mathcal{X}^{\text{bin}},v,\theta} (15) \text{ s.t. } (13), (18)-(33). \qquad \text{(L AC-UC)}$$

In the DC-UC problem, we neglect reactive power and line losses, and voltage magnitudes deviations. It then follows that the real and apparent power flows are:

$$p^{\text{ft}}_{\ell,t} = -p^{\text{tf}}_{\ell,t} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T} \qquad (39)$$

$$p^{\text{ft}}_{\ell,t} = -B_{\ell}(\theta_{\ell_{\text{f}}} - \theta_{\ell_{\text{t}}}) \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T} \qquad (40)$$

$$s^{\text{ft}}_{\ell,t} = p^{\text{ft}}_{\ell,t}, \; s^{\text{tf}}_{\ell,t} = -p^{\text{ft}}_{\ell,t} \qquad \forall \ell \in \mathcal{L}, t \in \mathcal{T}. \qquad (41)$$

Now, we define the DC-UC problem as:

$$\min_{\mathcal{X}^{\text{cont}},\mathcal{X}^{\text{bin}},\theta,p^{\text{ft}},p^{\text{tf}}} (15) \quad \text{s.t. } (18)-(27), (29), (31), (33), (34), (39)-(41). \qquad \text{(DC-UC)}$$

F. Multi-Time Period AC-OPF Formulation

When formulating the MTP AC-OPF problem, we assume all elements in $\mathcal{X}^{\text{bin}}$ are known parameters, i.e., the commitment schedule is set, and we optimize over continuous generation variables $\mathcal{X}^{\text{cont}}$. The MTP AC-OPF problem follows:

$$\min_{\mathcal{X}^{\text{cont}},\theta,v,p^{\text{ft}},p^{\text{tf}},q^{\text{ft}},q^{\text{tf}}} (15) \text{ s.t. } (18)-(38). \qquad \text{(MTP AC-OPF)}$$

The MTP AC-OPF problem is used to test the feasibility of the commitment schedules resulting from solving the NN AC-UC, L AC-UC, and DC-UC problems. We classify a UC solution that results in an infeasible MTP AC-OPF problem as a *infeasible commitment schedule*, i.e., the selection of binary variables in set $\mathcal{X}^{\text{bin}}$ cannot be realized.

IV. Test Results

In this section, we present results collected on the 14-, 57-, and 89-bus PGLib-OPF test cases [29] over a 24-hour period. In Section IV-A, we first compare the expressive power of the compact NN model (9) to a direct power flow mapping, which is exclusively used in the literature. In Section IV-B, we use the compact NN models to solve the NN AC-UC problems; then, we compare the obtained solutions to those generated by the linear benchmarks.

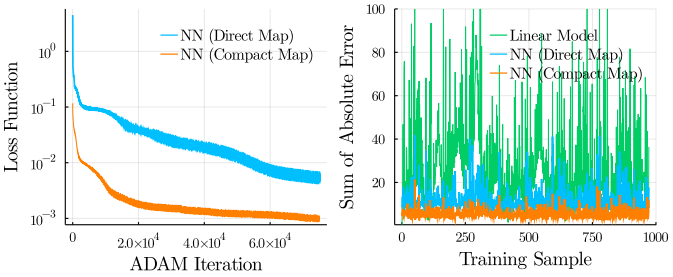

Figure 1. The left panel shows the loss function of the compact the direct NN mappings during training. The right panel shows linear, direct NN, and compact NN per-unitized error ($\|y_{\text{pf}} - y_{\text{lin}}\|_1$, $\|y_{\text{pf}} - y_{\text{nn}}\|_1$, and $\|y_{\text{pf}} - y_{\text{pw}}\|_1$, respectively) associated with the the 89-bus system.

A. Expressive Power of the Compact NN

In order to test the expressive power of the compact NN, we collected 972 feasible power flow solutions from the 89-bus system; loads were chosen by looping over UC load curves (see the following subsection for more details regrading data collection). We then trained a compact NN power flow mapping of the form (9) with $\rho = 25$ ReLUs using ADAM in Flux. Data were shuffled and mini-batched into sets of 75, and a learning rate of $\eta = 2.5 \times 10^{-4}$ was used. A second NN was also trained, but this model mapped power flow inputs (v and θ) *directly* to power flow outputs via $y_{\text{nn}} = \boldsymbol{w}_2 \sigma(\boldsymbol{w}_1^T x + b)$, which we refer to as a "direct" NN mapping; this model was also trained with 25 ReLUs. Both NNs were allowed to train for 7.5×10^4 steps, where loss function minimization showed signs of saturation. Results are shown in Figure 1, where the left panel shows loss function saturation, and the right panel depicts the predictive accuracy of the models. For reference, we also plot the prediction of linear power flow model $y_{\text{lin}} = \boldsymbol{J}^{\star} x + r^{\star}$ from (13). NN power flow predictions are generally an order of magnitude better than linear model predictions, and the compact NN predictions are generally over a factor of two better than the direct NN.

B. Data Collection and NN Training

We used the UC nodal load curves and generation cost curves developed in the UnitCommitment.jl package [30] for these systems. Reactive power load curves were generated for each system by assuming constant power factors at each load. For increased complexity, we assumed the active power limits for each generator in the 14 and 57 bus systems matched those of the associated MATPOWER test cases (i.e., contrary to PGLib-OPF, MATPOWER assumes that no generating unit acts only as a synchronous condenser; instead, every generator can produce active power and, therefore, a larger number of generators can participate in UC). We also decreased the apparent power thermal limits in these systems by 30%.

The power flow mapping (9) can potentially be trained on any set of feasible power flow solutions. To collect training data in a targeted way, we first gathered the hourly load profiles associated with the UC problems. For each hour, we used PowerModels.jl [26] to generate a *feasible* power flow solution (i.e., a power flow solution which satisfied all

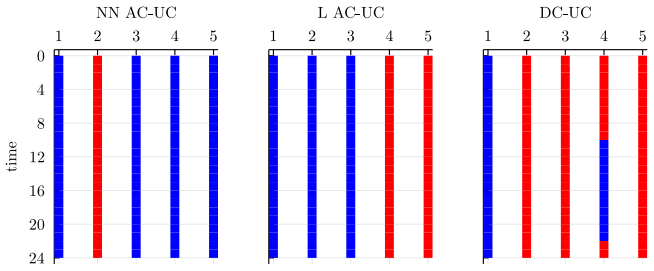

Figure 2. These three plots show the unit status decisions for the 14-bus network resulting from solving the NN AC-UC, L AC-UC and DC-UC problems. The top labels $(1-5)$ refer to the unit numbers, and the left axis marks a 24-hour period. Blue sections represent time periods where the status of the corresponding unit is ON, while red sections represent OFF periods.

thermal, voltage, and generation limit constraints). For each hour, we also generated feasible power flow solutions with each generator turned off, and with random combinations of up to three other generators also turned off; infeasible samples were rejected. In order to sample over a larger feasible space, we also perturbed the generator voltage limits at each power flow solve. That is, we randomly "pushed" the $V^{\min}$ and $V^{\max}$ generator constraints up and down, respectively, to nonstandard values which were still within the feasible space.

Once training and testing data sets were collected, compact NN-based power flow mapping models were trained and compressed. NN models of the 14-, 57-, and 89-bus power systems were trained on 219, 532, and 972 power flow samples, respectively. Notably, each training sample contained $2n-1$ inputs (reference bus phase angle was never included as an input) and $2n+2m$ outputs, according to mapping (4). The NNs associated with these systems contained 20, 25, and 30 ReLUS, respectively. Models were trained using ADAM in Flux, learning rates were set between $(1-3)\times 10^{-4}$, and all data were shuffled and mini-batched.

C. Unit Commitment Experiment Results

We solved the three versions of the UC problem discussed in Section III (NN AC-UC, L AC-UC, and DC-UC) for the three considered networks (14-, 57-, and 89-bus). For each system, the linearization terms $(\boldsymbol{J}^\star, r^\star)$ were generated from a power flow solution of the UC hour-1 (mean value) base load level with all generators turned ON and producing. Figure 2 shows the resulting unit commitment decisions for the 14-bus network. Blue and red lines mark time periods when the corresponding unit is ON and OFF, respectively. Here, we clearly see different unit commitment schedules are chosen. Note that the NN-based solution turns on more generators than the other power flow approximations. Across the three tested networks, the commitment schedules resulting from the NN-based and linear formulations are MTP AC-OPF feasible (see Section III-F) in the base-loading case. However, the committment schedules found via DC-UC only resulted in a feasible MTP AC-OPF for the 57-bus network.

Next, we tested the performance of the UC formulations using three different load alteration schemes with the goal of capturing various multi-time loading possibilities. In the first scheme, we uniformly scaled all real and reactive loads in the network over all time periods using the same scaling value. In the second scheme, we scaled the loads at each bus over all time periods using a randomly sampled value. The last load alteration scheme aimed to increase the peak loads and decrease the lowest loads, thus forcing more time-varying commitment decisions. We accomplished this by scaling each load at each hour t according to: $1+a\sin(\frac{2\pi}{24t})$, where a is the amplitude of the sine. We tested 10 cases for each loading scheme, for a total of 30 varied loading scenarios, and constrained our scaling range to $\pm 15\%$ of the original loads. For the first and last loading schemes, we tested samples evenly distributed within this scaling range.

Table I summarizes the MTP AC-OPF feasibility results, which use the binary variables from the UC solutions shown in the left-most column. Each UC formulation has a tallied number of total feasible and infeasible solutions, as well as the number of scenarios that did not produce a solution (either due to an infeasible UC problem or the inability of the MTP AC-OPF problem to converge within a reasonable time limit). The UC problems were run until a relative MIP gap of 1% was achieved; otherwise, the best feasible solution found after one hour of solving was used. If a MTP AC-OPF feasible solution was found for the unit commitment decisions resulting from the NN AC-UC problem and for either the L AC-UC or DC-UC problems, the resulting total operation costs from these simulations were similar. That is, across all considered networks and all loading scenarios, the maximum difference between two feasible solutions for a given network and loading profile was 0.02%. Therefore, we focus our discussion on feasibility rather than optimality.

Overall, the NN-based method outperforms the linear and DC approximations. There are multiple cases where the NN-based approximation is the *only* formulation that selects a feasible unit commitment schedule. For the 14-bus network, this is true for 15 out the the total 30 test cases, and for the 89-bus network, 5 out of the 30 test cases. Furthermore, the NN-based commitment schedules are *only* MTP AC-OPF infeasible (or unable to find a solution) when both the linear and DC methods are infeasible as well. Also, note that the feasibility of the test cases were not verified.

For the 57-bus network, we found that the linear power-flow approximation performed equally as well as the NN-based approximation in all cases. Most unit commitment schedules for this test case required all generators to be ON at all times, which, we hypothesize, did not require the additional accuracy afforded by the NN-based UC formation.

Lastly, we compared the apparent power flows predicted in the NN-based and linear UC problems to those of the actual apparent power flows calculated via their associated MTP AC-OPFs. Figure 3 compares the 1-norm error between the predicted and actual apparent power flows for the NN-based (red dots) and linear (blue dots) methods for the 30 considered loading cases in Table I. The NN-based approximation outperforms the linear approximation in all but one case. Apparent power flows in the opposite direction (s^{tf}) has similar results to those shown in Figure 3.

TABLE I
MTP AC-OPF RESULTS FOR LOAD VARIATIONS

		14-bus	57-bus	89-bus
NN AC-UC	Feasible	28	30	28
	Infeasible	2	0	1
	No solution	0	0	1
L AC-UC	Feasible	13	30	22
	Infeasible	15	0	8
	No solution	2	0	0
DC-UC	Feasible	0	23	19
	Infeasible	30	7	10
	No solution	0	0	1

V. CONCLUSIONS

This paper has demonstrated proof-of-concept for modeling AC power flow constraints with a compact NN-based piecewise linear power flow mapping; this mapping was learned directly from feasible power flow solutions. Once trained, we replaced the power flow constraints inside of the AC-UC problem with the NN-based model. Hence, the AC-UC MINLP is transformed into a more tractable MILP. Our results show that the NN-based formulation often generates feasible commitment schedules when the benchmark models (DC-UC and L AC-UC) could not. Furthermore, the NN-based formulation only produced infeasible schedules when both benchmark models did as well. These results confirm that low-rank updates of a linear power flow model can successfully approximate the nonlinear power flow equations across expansive operational regions, and they show that NN-based modeling of power flow can help alleviate the computational burden of the AC-UC problem in way that increases the feasibility of scheduling decisions relative to linear models.

While embedding the learned power flow mapping into the UC problem generally resulted in a greater number of feasible commitment decisions, the computational cost of the associated UC problem also increased. A NN with ρ ReLU activation functions applied to a UC problem spanning T hours will result in the addition of $\rho \times T$ new binary decision variables. In this paper, we took steps to both limit ρ and increase the sparsity of the generally dense NN weighting matrices. Future work, however, will investigate other methods for limiting computational expense, e.g., by parameterizing which ReLU activation function can be active as a function of the time varying load profile: $\rho(T)$. On top of the computational expense challenge, we also found that the performance of the NN depended strongly on the power flow samples used to train the model. Choosing power flow training data samples which were not representative of the UC problem negatively impacted the feasibility of commitment decisions in testing. In this paper, we utilized a simplistic rejection sampling approach for collecting feasible power flow training data. Future work will utilize more advanced, optimization-based methods for collecting representative training data across more targeted regions of power flow space.

Finally, in this paper, the performance of the NN model was evaluated both statistically (i.e., by computing model performance on sampled testing data) and through UC commitment decisions (i.e., schedule feasibility). In order to engender stronger trust in the learned model across larger operational regions, future work will focus on using optimization-based methods to rigorously verify the performance of the NN model.

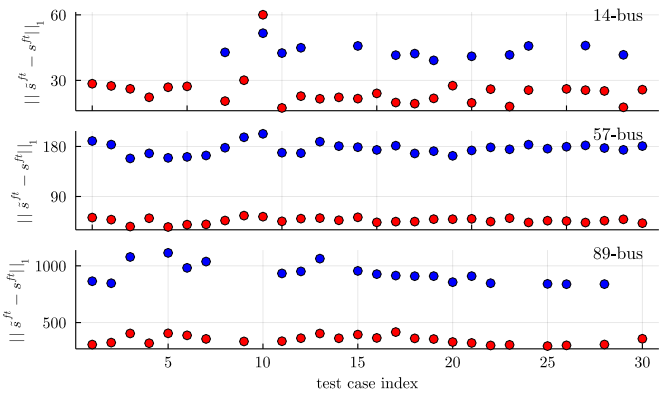

Figure 3. These three plots show the 1-norm error between the predicted apparent power flows ($\tilde{s}^{\text{ft}}$, calculated via the NN AC-UC and L AC-UC) and the actual apparent flows (s^{ft}, calculated via the MTP AC-OPF) in per-unit for the 30 test cases with varied loads. Red dots mark the 1-norm error associated with the NN-based power flow approximation, and blue dots mark the linear approximation. Missing data points correspond to infeasible solutions or problems that did not find a solution within a reasonable time.

APPENDIX

The power injection Jacobian $\boldsymbol{J}_{pq} \in \mathbb{R}^{2n\times 2n}$ relates polar voltage (v, θ) and nodal power injection (p^{inj}, q^{inj}) perturbations via

$$\boldsymbol{J}_{pq} = (\langle \mathtt{d}(\boldsymbol{Y}_b v e^{j\theta})^* \rangle + \langle \mathtt{d}(v e^{j\theta}) \rangle N \langle \boldsymbol{Y}_b \rangle) R(v e^{j\theta}), \qquad (42)$$

where $\mathtt{d}(\cdot)$ is the diagonalization operator, and $R(\cdot)$, N, and $\langle\cdot\rangle$ are given in [31]. The apparent power line flow Jacobians, $\boldsymbol{J}_{s,\text{ft}}, \boldsymbol{J}_{s,\text{tf}}, \in \mathbb{R}^{m\times 2n}$, can be constructed by first partitioning the Jacobians relating active and reactive power flows in the lines:

$$\boldsymbol{J}_\gamma = (\langle \mathtt{d}(\boldsymbol{Y}_\gamma v e^{j\theta})^* \boldsymbol{E}_\gamma \rangle + \langle \mathtt{d}(\boldsymbol{E}_\gamma v e^{j\theta}) \rangle N \langle \boldsymbol{Y}_\gamma \rangle) R_v \qquad (43a)$$

$$= \begin{bmatrix} \frac{\partial p^\gamma}{\partial v} & \frac{\partial p^\gamma}{\partial \theta} \\ \frac{\partial q^\gamma}{\partial v} & \frac{\partial q^\gamma}{\partial \theta} \end{bmatrix}, \quad \gamma \in \{\text{ft}, \text{tf}\} \qquad (43b)$$

where $R_v \triangleq R(v e^{j\theta})$. Since apparent power is related to active and reactive power via (37)-(38), the chain rule yields the Jacobian of s, where $x_{\mathtt{d}} \triangleq \mathtt{d}(x)$:

$$\boldsymbol{J}_{s,\gamma} = (s^\gamma_{\mathtt{d}})^{-1} \begin{bmatrix} p^\gamma_{\mathtt{d}} \frac{\partial p^\gamma}{\partial v} + q^\gamma_{\mathtt{d}} \frac{\partial q^\gamma}{\partial v} & p^\gamma_{\mathtt{d}} \frac{\partial p^\gamma}{\partial \theta} + q^\gamma_{\mathtt{d}} \frac{\partial q^\gamma}{\partial \theta} \end{bmatrix}. \qquad (44)$$

REFERENCES

[1] D. Bienstock and A. Verma, "Strong NP-hardness of AC power flows feasibility," *Operations Research Letters*, vol. 47, no. 6, pp. 494–501, 2019.

[2] K. Baker, "Solutions of DC OPF are never AC feasible," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 2021, pp. 264–268.

[3] J. D. Foster, "Mixed-integer quadratically-constrained programming, piecewise-linear approximation and error analysis with applications in power flow," dissertation, The University of Newcastle, Australia, School of Mathematical and Physical Sciences, November 2013.

[4] S. I. Nanou, G. N. Psarros, and S. A. Papathanassiou, "Network-constrained unit commitment with piecewise linear AC power flow constraints," *Electric Power Systems Research*, vol. 195, p. 107125, 2021.

[5] B. Donon, R. Clément, B. Donnot, A. Marot, I. Guyon, and M. Schoenauer, "Neural networks for power flow: Graph neural solver," *Electric Power Systems Research*, vol. 189, p. 106547, 2020.

[6] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2082–2092, 2021.

[7] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," *arXiv:1711.07356*, 2017.

[8] I. Murzakhanov, A. Venzke, G. S. Misyris, and S. Chatzivasileiadis, "Neural networks for encoding dynamic security-constrained optimal power flow," *arXiv:2003.07939*, Oct. 2021.

[9] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020.

[10] J. Katz, I. Pappas, S. Avraamidou, and E. N. Pistikopoulos, "The integration of explicit MPC and ReLU based neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11 350–11 355, 2020.

[11] J. Huchette, H. Lu, H. Esfandiari, and V. Mirrokni, "Contextual reserve price optimization in auctions via mixed-integer programming," in *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[12] B. Say, G. Wu, Y. Q. Zhou, and S. Sanner, "Nonlinear hybrid planning with deep net learned transition models and mixed-integer linear programming," in *26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 750–756.

[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] Y. Zhang, C. Chen, G. Liu, T. Hong, and F. Qiu, "Approximating trajectory constraints with machine learning–microgrid islanding with frequency constraints," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1239–1249, 2020.

[15] Y. Zhang, H. Cui, J. Liu, F. Qiu, T. Hong, R. Yao, and F. F. Li, "Encoding frequency constraints in preventive unit commitment using deep learning with region-of-interest active sampling," to appear in *IEEE Transactions on Power Systems*, 2022.

[16] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018.

[17] K. Y. Xiao, V. Tjeng, N. M. Shafiullah, and A. Madry, "Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability," *arXiv e-prints*, p. arXiv:1809.03008, Sep. 2018.

[18] R. Mishra, H. P. Gupta, and T. Dutta, "A survey on deep neural network compression: Challenges, overview, and solutions," *arXiv preprint arXiv:2010.03954*, 2020.

[19] C.-L. Tseng, *On power system generation unit commitment problems*. University of California, Berkeley, 1996.

[20] J. Liu, C. D. Laird, J. K. Scott, J.-P. Watson, and A. Castillo, "Global solution strategies for the network-constrained unit commitment problem with AC transmission constraints," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1139–1150, 2018.

[21] C. Murillo-Sanchez and R. J. Thomas, "Thermal unit commitment including optimal AC power flow constraints," in *31st Hawaii International Conference on System Sciences (HICSS)*, vol. 3, 1998, pp. 81–88.

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

[23] F. Zohrizadeh, M. Kheirandishfard, A. Nasir, and R. Madani, "Sequential relaxation of unit commitment with AC transmission constraints," in *IEEE Conference on Decision and Control (CDC)*, December 2018, pp. 2408–2413.

[24] B. Donon, B. Donnot, I. Guyon, and A. Marot, "Graph neural solver for power systems," in *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[25] A. Venzke and S. Chatzivasileiadis, "Verification of neural network behaviour: Formal guarantees for power system applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, 2021.

[26] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *20th Power Systems Computation Conference (PSCC)*, 2018.

[27] D. Shchetinin, T. T. De Rubira, and G. Hug, "Efficient bound tightening techniques for convex relaxations of AC optimal power flow," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3848–3857, 2019.

[28] G. Morales-España, J. M. Latorre, and A. Ramos, "Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1288–1296, 2012.

[29] S. Babaeinejadsarookolaee, A. Birchfield *et al.*, "The power grid library for benchmarking AC optimal power flow algorithms," *arXiv:1908.02788*, Aug. 2019.

[30] A. S. Xavier, A. M. Kazachkov, and F. Qiu, "ANL-CEEESA/UnitCommitment.jl: v0.2.2," Jul. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5120043

[31] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 402–409.