Unsupervised Learning for Solving AC Optimal Power Flows: Design, Analysis, and Experiment

Wanjun Huang, *Member*, *IEEE*, Minghua Chen, *Fellow*, *IEEE*, and Steven H. Low, *Fellow*, *IEEE*

***Abstract*—With the increasing penetration of renewables, AC optimal power flow (AC-OPF) problems need to be solved more frequently for reliable and economic power system operation. Supervised learning approaches have been developed to solve AC-OPF problems fast and accurately. However, due to the non-convexity of AC-OPF problems, it is non-trivial and computationally expensive to prepare a large training dataset, and multiple load-solution mappings may exist to impair learning even if the dataset is available. In this paper, we develop an *unsupervised* learning approach (DeepOPF-NGT) that does not require ground truths. DeepOPF-NGT utilizes a properly designed loss function to guide neural networks in directly learning a legitimate load-solution mapping. Kron reduction is used to remove the zero-injection buses from the prediction. To tackle the unbalanced gradient pathologies known to deteriorate the learning performance, we develop an adaptive learning rate algorithm to dynamically balance the gradient contributions from different loss terms during training. Further, we derive conditions for unsupervised learning to learn a legitimate load-solution mapping and avoid the multiple mapping issue in supervised learning. Results of the 39/118/300/1354-bus systems show that DeepOPF-NGT achieves optimality, feasibility, and speedup performance comparable to the state-of-the-art supervised approaches and better than the unsupervised ones, and a few ground truths can further improve its performance.**

***Index Terms*—AC optimal power flow, unsupervised learning, deep neural network, adaptive learning rate, Kron reduction.**

I. Introduction

The essential AC optimal power flow (AC-OPF) problem has been extensively investigated to minimize operational costs by optimizing power generation subject to physical and operational constraints. A study conducted by FERC highlights the potential of efficient AC-OPF solutions, which could result in substantial annual savings amounting to tens of billions of dollars [1]. With the increasing penetration of renewable energy sources such as wind and solar power, the AC-OPF problem will need to be solved more frequently to maintain the stable and economic operation of power systems. As a result, it is of great interest to solve AC-OPF problems with high efficiency.

Wanjun Huang is with the School of Automation Science and Electrical Engineering, Beihang University. Minghua Chen is with the School of Data Science, City University of Hong Kong. Steven H. Low is with the Department of Computing & Mathematical Sciences and the Department of Electrical Engineering at Caltech and the University of Melbourne, Australia.

This work is supported in part by a General Research Fund from Research Grants Council, Hong Kong (Project No. 11200223), an InnoHK initiative, The Government of the HKSAR, Laboratory for AI-Powered Financial Technologies, and a Shenzhen-Hong Kong-Macau Science & Technology Project (Category C, Project No. SGDX20220530111203026). The authors would also like to thank the anonymous reviewers for their helpful comments. Corresponding author: Minghua Chen (minghua.chen@cityu.edu.hk).

In recent decades, considerable advancements have been made in optimizing techniques for efficient AC-OPF problem-solving. These techniques can be broadly classified into mathematical and metaheuristic methods. Conventional mathematical methods mainly contain the gradient method, Newton method, linear programming, quadratic programming, and Interior Point method [2]. Additionally, decomposition algorithms such as Benders decomposition and alternating direction method of multipliers have been proposed to reduce problem size [2]. However, these methods encounter inherent limitations in computational speedup due to the need for repetitive solving of AC-OPF problems for the same system but with varying loads. To achieve faster computational speeds, metaheuristic techniques such as genetic algorithms and particle swarm optimization have been introduced [3]. Nevertheless, these methods still face significant computational burdens and scalability challenges when applied to large-scale systems.

Recently, machine learning methods have gained significant attention for their promising achievements in solving AC-OPF problems [4], demonstrating computational speedups of up to 2$\sim$3 orders of magnitude compared to conventional solvers [5], [6]. Most machine learning schemes employ supervised learning techniques to learn the load-solution mapping, utilizing datasets generated by physics-based solvers, such as the interior point optimizer (IPOPT) solver. Well-trained deep neural network (DNN) models have been utilized to reduce the number of constraints [7]–[11], expedite the convergence of conventional solvers [12]–[14], or directly predict decision variables [5], [15]–[24]. Further discussions on related works are presented in Section II.

Despite the promising results achieved, the supervised learning approach faces two primary limitations. First, it requires a large training dataset with ground truths to achieve considerable accuracy. However, obtaining such a dataset is non-trivial and computationally expensive as the AC-OPF problems are non-convex and hard to solve, particularly for large-scale instances [25]. Second, even if such a large dataset is available, DNNs may still fail to perform well. Due to the non-convexity of AC-OPF problems, existing solvers may provide one of the globally/locally optimal solutions, and different initial points may lead to diverse solutions [26]. Hence, the training data may encompass different load-(sub-)optimal-solution mappings. As a result, the supervised learning method may learn an average of these mappings that exhibit inferior performance; see Section III-B for an illustrating example. It is challenging to prepare a dataset containing ground truths from only one mapping. A study by Kotary et al. [27] propose to address this issue by a bi-level optimization approach. However, the problem

TABLE I
COMPARISON OF EXISTING MACHINE LEARNING BASED-APPROACHES FOR SOLVING OPF PROBLEMS

Category	Approach	Existing study	Problem		Metrics in consideration		Ground truth	Gradient pathologies
			DC-OPF	AC-OPF	Feasibility	Optimality		
Supervised learning	Predict active/ inactive constraints	[7], [8]	✓		✗	✓	✓	✗
		[9], [10]	✓		✓	✓	✓	✗
		[11]	✓	✓	✗	✓	✓	✗
	Predict warm-start points	[12]	✓		✓	✓	✓	✗
		[13], [14]		✓	✓	✓	✓	✗
	Learn load-solution mapping	[15]–[19]	✓		✓	✓	✓	✗
		[20], [21], [29]		✓	✗	✓	✓	✗
		[5], [6], [30]		✓	✓	✓	✓	✗
Reinforcement learning	Learn load-solution mapping	[22]–[24]		✓	✓	✓	✗	✗
Unsupervised learning	Learn load-feasible-solution mapping and make refinements	[31]		✓	✓	✓	✓	✗
	Learn load-solution mapping	[32]–[35]		✓	✓	✓	✗	✗
		This work		✓	✓	✓	✗	✓

is proven NP-hard, and finding exact solutions for large-scale instances is computationally prohibitive. A generative learning approach is developed to solve non-convex problems with multi-valued input-solution mapping in [28]. It would be interesting to investigate its application to the non-convex AC-OPF setting.

To tackle the aforementioned issues, we propose an *unsupervised* learning approach called DeepOPF-NGT (where NGT denotes "No Ground Truth") to solve AC-OPF problems efficiently. Unsupervised learning, as we know, focuses on extracting patterns and relationships from unlabeled data, without using explicit labels or predefined outcomes. DeepOPF-NGT is designed to discern the mapping between load and solution without the need for ground truth, thereby circumventing learning the average of multiple mappings - a scenario that could lead to subpar performance due to the existence of multiple global or local optimal solutions for each input. Remarkably, we demonstrate that DeepOPF-NGT can learn a legitimate mapping (one that produces a unique global or local optimal solution for each input) under certain conditions. Although reinforcement learning also does not require ground truth, it is primarily developed to tackle multi-step or sequential decision-making problems in uncertain and dynamic environments, while this study focuses on the standard AC-OPF problem, a single-step problem within certain scenarios. Detailed discussion is provided in Section II. Our main contributions are three-fold:

- An *unsupervised* learning approach, DeepOPF-NGT, is proposed to solve AC-OPF problems without the need for ground truths. It leverages a properly designed loss function consisting of the objective and the penalties for constraint violations and dissatisfied loads to guide DNNs to learn a legitimate load-solution mapping directly. Kron reduction [36] is employed to remove the zero-injection buses (ZIBs) from the prediction, improving the scalability and feasibility.
- An adaptive learning rate algorithm is designed to tackle the unbalanced gradient pathologies known to deteriorate the learning performance [37] by dynamically balancing the gradient contributions from different terms in the loss function. Besides, we provide theoretical insights into learning a legitimate mapping for optimization problems with multiple mappings by the *unsupervised* learning method. To the best of our knowledge, this gives the first justification of applying *unsupervised* learning to learn a legitimate mapping and avoid the multiple mapping issue encountered in supervised learning (see Section III-B for an illustrating example). Moreover, we consider the situation with a few ground truths and leverage them to improve DeepOPF-NGT by designing a semi-supervised learning approach.
- Simulation results of the modified IEEE 39/118/300/1354 bus systems validate that DeepOPF-NGT achieves comparable performances to state-of-the-art supervised and *unsupervised* learning approaches when there is only one global/local optimal solution for each load input, but outperforms these approaches when there are multiple global/local optimal solutions for each load input. Besides, it has good scalability. Furthermore, the proposed adaptive learning rate algorithm can alleviate the unbalanced gradient pathologies, and a few ground truths can improve DeepOPF-NGT.

The structure of the remaining part is as follows. Section II discusses the related works. Section III formulates the AC-OPF problem and explains the multiple load-solution mapping issue. Section IV illustrates the framework of DeepOPF-NGT. Section V analyzes the performance of DeepOPF-NGT and extends it to a semi-supervised learning approach. Section VI validates the effectiveness of DeepOPF-NGT. Section VII concludes this paper and discusses future directions.

II. RELATED WORK

The field of machine learning offers three main categories for addressing optimal power flow (OPF) problems: supervised learning, unsupervised learning, and reinforcement learning, as outlined in Table I.

Within the supervised learning approach, two distinct subcategories can be identified: hybrid and stand-alone approaches. The hybrid approach involves predicting active [8], [11] or inactive [7], [9], [10] constraints to reduce the problem size or warm-start points to expedite the convergence of conventional solvers [12]–[14]. To enhance solution feasibility, some studies incorporate penalties for constraint violations into the loss function [9], [10], [12]–[14]. Nevertheless, the potential for

computational speedup is limited, as the AC-OPF problem still needs to be solved iteratively.

The stand-alone approach directly predicts solutions for OPF problems. Several researchers utilize DNNs to predict all variables and incorporate constraint violations into the loss function using Lagrangian duality, aiming to enhance feasibility [20], [21], [29]. By avoiding the need to solve the optimization problem, this approach achieves a higher computational speedup compared to the hybrid approach. Nonetheless, the resulting solution is prone to in-feasibility due to prediction errors, as it does not guarantee the satisfaction of equality constraints. To tackle this challenge, several studies [15]–[17] propose a two-stage predict-and-reconstruct framework to solve the (security-constrained) DC-OPF problem. This approach predicts a subset of the variables and reconstructs the remaining ones by leveraging the equality constraints. The study in [5] extends this framework to the AC-OPF problem, which predicts decision variables and reconstructs the remaining variables using a power flow solver. However, due to the requirement of solving power balance equations, the computational speedup of this extended framework remains limited. Furthermore, the presence of prediction errors can lead to scenarios with no feasible solution or voltage violations. To simultaneously enhance computational speedup and feasibility, DeepOPF-V predicts all bus voltages and reconstructs the remaining variables via simple matrix operations [6]. Topology-adaptive approaches have also emerged to handle grid topology changes [38], [39]. Moreover, a recent study in [30] proposed the low complexity homeomorphic projection to ensure solution feasibility.

In contrast to the supervised learning approach that learns the load-solution mapping embedded in the training dataset, the unsupervised learning approach utilizes the loss function to guide DNNs in learning the load-solution mapping directly. However, only a few unsupervised learning approaches have been proposed for solving OPF problems. In [32], a Deep Constraint Completion and Correction (DC3) algorithm is developed to enforce feasibility through a differentiable procedure designed for optimization problems with hard constraints. Meanwhile, the proposed DeepOPF-NGT significantly expands our preliminary work in [33][1] with three critical improvements: (i) a substantial improvement in training efficiency is achieved through the application of Kron reduction [36] and the proposed adaptive learning rate algorithm, which is particularly essential for large-scale systems[2]; (ii) the multiple mapping issue of the AC-OPF problem is explored, offering theoretical insights into learning one legitimate mapping; (iii) DeepOPF-NGT is extended to a semi-supervised learning approach.

Noticeably, DeepOPF-NGT differs from DC3 [32] in several key aspects. First, DeepOPF-NGT achieves much faster training speeds compared to DC3. While DC3 learns the "load-generation" mapping and solves power flow equations to reconstruct the remaining variables (a similar approach was presented in [16]), DeepOPF-NGT learns the "load-voltage" mapping and reconstructs the remaining variables through simple matrix operations. Second, the training process of DeepOPF-NGT is simpler. In DC3, the gradient of the loss function is implicit, and computing the inverse of the Jacobian matrix can be time-consuming and prone to numerical issues (as discussed in Section VI-B). In contrast, DeepOPF-NGT explicitly computes the gradient of the loss function and only requires the computation of the Jacobian matrix. Third, DC3 employs fixed coefficients for different terms in the loss function, whereas DeepOPF-NGT devises an adaptive learning rate algorithm to address gradient pathologies [37]. Moreover, in contrast to DC3 which was evaluated on a 57-bus system, DeepOPF-NGT has been validated on larger power systems, i.e., IEEE 118/300/1354-bus systems. Fourth, we prove for the first time in the literature that the *unsupervised* learning method can guarantee learning a legitimate mapping under certain conditions. Fifth, we extend DeepOPF-NGT to a semi-supervised learning approach.

Furthermore, we have noted recent studies on *unsupervised* learning methods in [31], [34], [35]. However, the approach presented in [31] does not strictly adhere to the definition of *unsupervised* learning, as it still requires a large training dataset containing feasible solutions. Besides, it approximates the solution feasibility using DNNs, which can introduce potential inaccuracies. Similar to [32] and [33], the approaches developed in [34], [35] utilize a loss function consisting of the objective and constraint violation penalties. However, these methods have limited speedup for solving power flow equations and do not consider the unbalanced gradient pathologies that are known to deteriorate the performance of *unsupervised* learning. Additionally, the approach in [35] needs to solve dual problems at each training iteration and has only been validated for scenarios with 10% load variations, whereas our proposed DeepOPF-NGT does not require solving optimization problems and has been validated in scenarios with over 40% load variations.

Distinct from the supervised and unsupervised learning approaches that primarily address single-step problems within certain scenarios, reinforcement learning aims to tackle multi-step problems in an uncertain and dynamic environment. Similar to unsupervised learning, it operates independently of ground truths. In reinforcement learning, an agent learns to make sequential decisions to maximize long-term rewards by interacting with the environment. The agent receives feedback from the environment depending on its previous states and learns through the trial and error technique. The study in [23] devises a reward function consisting of total generation costs and constraint violation penalties. To handle constraints, the studies in [22], [24] employ Lagrange multipliers to update the weights of penalties adaptively [22], [24]. However, since we focus on the single-step standard AC-OPF problem, the reinforcement learning approach is unsuitable for this study.

III. Problem Formulation

A. AC Optimal Power Flow Model

The AC-OPF problem aims to minimize the generation cost while satisfying physical and operational constraints, such as power balance, generator capacity, transmission line limits,

[1] This workshop does not publish proceedings, and submissions are non-archival. Submission to this workshop does not preclude future publication.

[2] The work in [33] only presents results in the IEEE 30-bus system which is much smaller than the 300/1354-bus systems considered in this study.

voltage limits, renewable energy generation, the control of transformers and shunt compensators, etc. Without loss of generality, we consider the basic setting, i.e., the standard AC-OPF model. This simplified model is commonly used in the literature such as [40] and can be formulated as follows:

$$\min \sum\nolimits_{i\in\mathcal{N}_G} C_i(P_{gi}), \tag{1a}$$

$$\text{s.t. } P_{ij} = G_{ij}V_i^2 - V_iV_j(B_{ij}\sin\theta_{ij} + G_{ij}\cos\theta_{ij}), \forall (i,j)\in\mathcal{E}, \tag{1b}$$

$$Q_{ij} = -B_{ij}V_i^2 - V_iV_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}), \forall (i,j)\in\mathcal{E}, \tag{1c}$$

$$P_i = \sum\nolimits_{(i,j)\in\mathcal{E}} P_{ij},\ Q_i = \sum\nolimits_{(i,j)\in\mathcal{E}} Q_{ij},\ \forall i\in\mathcal{N}, \tag{1d}$$

$$P_i = P_{gi} - P_{di},\quad Q_i = Q_{gi} - Q_{di},\ \forall i\in\mathcal{N}, \tag{1e}$$

$$\underline{P}_{gi} \le P_{gi} \le \overline{P}_{gi},\ \underline{Q}_{gi} \le Q_{gi} \le \overline{Q}_{gi},\ \forall i\in\mathcal{N}, \tag{1f}$$

$$\underline{V}_i \le V_i \le \overline{V}_i,\quad \forall i\in\mathcal{N}, \tag{1g}$$

$$\underline{\theta}_{ij} \le \theta_{ij} \le \overline{\theta}_{ij},\quad \forall (i,j)\in\mathcal{E}, \tag{1h}$$

$$P_{ij}^2 + Q_{ij}^2 \le \overline{S}_{ij}^2,\quad \forall (i,j)\in\mathcal{E}, \tag{1i}$$

where $\mathcal{N}$ is the set of all buses which consist of generation buses (collected in the set $\mathcal{N}_G$) and load buses (collected in the set $\mathcal{N}_L$); $\mathcal{E}$ denote the set of branches; the branch from bus i to bus j is denoted as (i,j); g_{ij} and b_{ij} are conductance and susceptance of the branch (i,j), respectively; (P_i, Q_i) are the net active and reactive power injections, (P_{gi}, Q_{gi}) are the active and reactive power generations, (P_{di}, Q_{di}) are the active and reactive loads, (V_i, θ_i) are the voltage magnitude and angle, at bus i; $\theta_{ij} := \theta_i - \theta_j$ is the angle difference, and (P_{ij}, Q_{ij}) are the active and reactive branch power flows, at branch (i,j). The given upper and lower bounds of a variable x are represented by $\overline{x}$ and $\underline{x}$, respectively. This model minimizes the generation cost in (1a) subject to physical and operational constraints in (1b)-(1i). The branch power flows are given by (1b)-(1c). Kirchhoff's circuit laws are ensured by (1d). The net power injections are given by (1e). Inequalities (1f)-(1i) impose the operational limits for power generations, voltage magnitudes and angles, and branch power flows.

For ZIBs, equations (1b)-(1e) can be expressed in Cartesian coordinates as below:

$$\sum\nolimits_{j\in\mathcal{N}} (G_{ij}e_j - B_{ij}f_j) = 0,\quad \forall i\in\mathcal{N}_Z, \tag{2a}$$

$$\sum\nolimits_{j\in\mathcal{N}} (G_{ij}f_j + B_{ij}e_j) = 0,\quad \forall i\in\mathcal{N}_Z, \tag{2b}$$

where $e_j = V_j\cos\theta_j$ and $f_j = V_j\sin\theta_j$, and $\mathcal{N}_Z$ is the set of ZIBs regarded as internal buses in Kron reduction. By equation (2), we can remove the ZIBs from the AC-OPF model in (1).

DeepOPF-NGT is a general unsupervised learning approach that can be extended to more complex AC-OPF models with renewable energy sources, discrete control variables, topological changes, and contingencies by integrating the related constraints and objectives into the formulation (1). Furthermore, our framework is capable of seamlessly incorporating reinforcement learning techniques. This is achieved by introducing time-coupling constraints - like ramp-rate constraints for generators - into the AC-OPF model. Subsequently, the loss function delineated in Section IV-A is utilized as the action value or reward function.

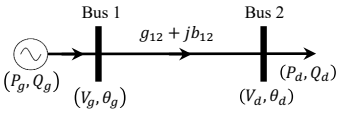


Fig. 1. An illustrating two-bus system: (P_g, Q_g) are active and reactive power generations, (P_d, Q_d) are active and reactive loads, and (V_g, θ_g) and (V_d, θ_d) are voltages of generation and load buses, respectively.

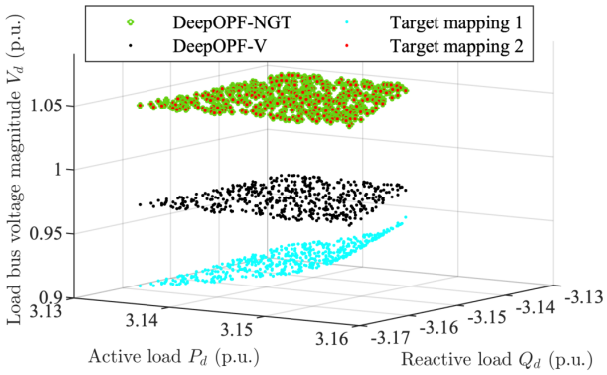


Fig. 2. Load-solution mappings in an illustrating two-bus system.

B. Multiple Mappings for the AC-OPF Problem

The AC-OPF problem in (1) may have multiple local or global optima due to its non-convex nature. Existing OPF solvers such as IPOPT may provide different local or global optimal solutions when starting from different initial points, resulting in mixed samples from different load-solution mappings. To illustrate this, we consider a two-bus system with one generation bus and one load bus (see Fig. 1) and use the IPOPT solver to generate 1000 training samples and 500 test samples with different load inputs. The method in [41] is used to find multiple global/ local solutions. For each load input, we generate 2000 different initial points within the bounds of each variable using uniform sampling and solve the AC-OPF problem in (1) starting from each initial point. The branch resistance is set to a small value such that the generation costs of different solutions are almost the same.

Fig. 2 displays the test samples, showcasing the mapping between loads and voltage magnitudes. There are two solutions with almost the same costs for each load input, forming two legitimate load-solution mappings, i.e., target mappings 1 and 2 marked by blue and red dots, respectively. When using the supervised learning approach, samples generated by the IPOPT solver may come from both target mappings, resulting in learning neither of the mappings. However, it is in principle not a concern for the *unsupervised* learning that does not require ground truths to operate.

We compared DeepOPF-NGT with the supervised learning approach DeepOPF-V [6] in learning the load-solution mapping using the aforementioned dataset. Due to space limitations, we only show the predicted load bus voltage magnitude in Fig. 2. The mappings learned by DeepOPF-NGT and DeepOPF-V are marked by green circles and black dots, respectively. The mapping learned by DeepOPF-NGT aligns almost perfectly

with target mapping 2 (which has slightly smaller costs than target mapping 1), satisfying all constraints and 99.85% active loads. Whereas, the mapping learned by DeepOPF-V is located between the two target mappings, satisfying all constraints but only 98.8% active loads. This example suggests that the *unsupervised* learning approach outperforms the *supervised* learning approach in the regression task with multiple load-solution correspondences for each load input.

IV. DeepOPF-NGT: An Unsupervised Learning Approach without Ground Truth

A. Schematic of DeepOPF-NGT

The schematic of the proposed DeepOPF-NGT is shown in Fig. 3. The DNN-based model is composed of fully connected layers, with a rectified linear unit (ReLU) activation function on each hidden layer and a sigmoid activation function on the output layer. It aims to learn the mapping between load configurations $(\boldsymbol{P_d}, \boldsymbol{Q_d})$ and nonzero-injection bus voltages $(\boldsymbol{V_\alpha}, \boldsymbol{\theta_\alpha})$, where $\boldsymbol{P_d}$ and $\boldsymbol{Q_d}$ are vectors of active and reactive loads, respectively, and $\boldsymbol{V_\alpha}$ and $\boldsymbol{\theta_\alpha}$ are vectors of voltage magnitudes and angles, respectively. Using the well-trained DNNs, the predicted voltage magnitudes $\hat{\boldsymbol{V}}_{\boldsymbol{\alpha}}$ and voltage angles $\hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}}$ can be obtained instantly with $(\boldsymbol{P_d}, \boldsymbol{Q_d})$ as the input. Next, the ZIB voltages $(\hat{\boldsymbol{V}}_{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{\beta}})$ are calculated by linear equation (2). Following that, based on the predicted voltages $(\hat{\boldsymbol{V}}, \hat{\boldsymbol{\theta}})$ and loads $(\boldsymbol{P_d}, \boldsymbol{Q_d})$, we can easily compute the left-hand side of equations (1e). Then, the remaining solution variables $(\hat{\boldsymbol{P}}_{\boldsymbol{g}}, \hat{\boldsymbol{Q}}_{\boldsymbol{g}})$ and the auxiliary variables $(\hat{\boldsymbol{P}}_{\boldsymbol{d}}, \hat{\boldsymbol{Q}}_{\boldsymbol{d}})$ are directly calculated by (1e) using the obtained left-hand side values without solving non-linear power flow equations. Specifically, for each bus i: 1) if there are only generators or loads, its predicted active and reactive generations (i.e., $\hat{P}_{gi}$ and $\hat{Q}_{gi}$) or predicted active and reactive loads (i.e., $\hat{P}_{di}$ and $\hat{Q}_{di}$) are obtained directly; 2) if there are both generators and loads, $\hat{P}_{di}$ and $\hat{Q}_{di}$ are set to the given loads P_{di} and Q_{di}, respectively, and then $\hat{P}_{gi}$ and $\hat{Q}_{gi}$ are directly calculated from (1e). The objective function is calculated by (1a) after obtaining $\hat{\boldsymbol{P}}_{\boldsymbol{g}}$. The post-processing method in [6] is used to improve the feasibility of the predicted solution. A PV bus will be switched to a PQ bus if the reactive power generation constraint is violated.

The loss function is designed as

$$\mathcal{L} = k_{obj}\mathcal{L}_{obj} + \mathcal{L}_{cons} + k_d\mathcal{L}_d, \tag{3}$$

where k_{obj} and k_d are positive constants, $\mathcal{L}_{obj}$ is the objective in (1a), $\mathcal{L}_{cons}$ is designed to find feasible solutions satisfying the constraints in (1f)-(1i), and $\mathcal{L}_d$ is designed to satisfy demanded loads. Kirchhoff's circuit laws in (1d) are satisfied automatically, since the net power injections can always be calculated with the predicted bus voltages. Specifically, $\mathcal{L}_{cons}$ is the penalty for constraint violations during training as below:

$$\mathcal{L}_{cons} = k_g\mathcal{L}_g + k_{S_l}\mathcal{L}_{S_l} + k_{\theta_l}\mathcal{L}_{\theta_l} + k_z\mathcal{L}_z, \tag{4}$$

where k_g, k_{S_l}, k_{θ_l} and k_z are positive constants; $\mathcal{L}_g$, $\mathcal{L}_{S_l}$, $\mathcal{L}_{\theta_l}$ and $\mathcal{L}_z$ are penalties for the violations of generation, branch flow, branch angle and ZIB voltage magnitude constraints during training, respectively, i.e.,

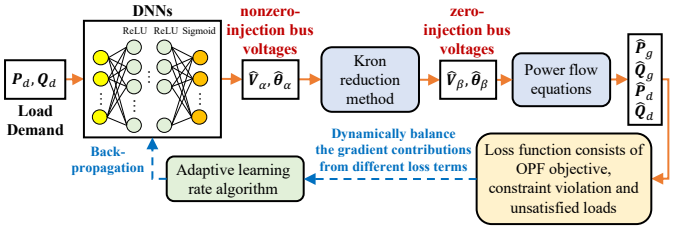


Fig. 3. Schematic of DeepOPF-NGT.

$$\mathcal{L}_g = \sum\nolimits_{i\in\mathcal{N}_G}[\max\left(\hat{P}_{gi} - \overline{P}_{gi}, 0\right)^2 + \max\left(\underline{P}_{gi} - \hat{P}_{gi}, 0\right)^2 + \max\left(\hat{Q}_{gi} - \overline{Q}_{gi}, 0\right)^2 + \max\left(\underline{Q}_{gi} - \hat{Q}_{gi}, 0\right)^2], \tag{5}$$

$$\mathcal{L}_{S_l} = \sum\nolimits_{(i,j)\in\mathcal{E}}[\max\left(\hat{S}_{ij} - \overline{S}_{ij}, 0\right)^2], \tag{6}$$

$$\mathcal{L}_{\theta_l} = \sum\nolimits_{(i,j)\in\mathcal{E}}[\max\left(\hat{\theta}_{ij} - \overline{\theta}_{ij}, 0\right)^2 + \max\left(\underline{\theta}_{ij} - \hat{\theta}_{ij}, 0\right)^2], \tag{7}$$

$$\mathcal{L}_z = \sum\nolimits_{i\in\mathcal{N}_Z}[\max\left(\hat{V}_i - \overline{V}_i, 0\right)^2 + \max\left(\underline{V}_i - \hat{V}_i, 0\right)^2]. \tag{8}$$

Here, $\hat{S}_{ij} = \sqrt{\hat{P}_{ij}^2 + \hat{Q}_{ij}^2}$, and $(\hat{P}_{ij}, \hat{Q}_{ij})$ are branch active and reactive power flows derived from predicted voltages $(\hat{\boldsymbol{V}}, \hat{\boldsymbol{\theta}})$. The term $\mathcal{L}_d$ penalizes the deviations between the demanded loads $(\boldsymbol{P_d}, \boldsymbol{Q_d})$ and the predicted loads $(\hat{\boldsymbol{P}}_{\boldsymbol{d}}, \hat{\boldsymbol{Q}}_{\boldsymbol{d}})$ as below:

$$\mathcal{L}_d = \sum\nolimits_{i\in\mathcal{N}_L/\mathcal{N}_Z}\left[(\hat{P}_{di} - P_{di})^2 + (\hat{Q}_{di} - Q_{di})^2\right]. \tag{9}$$

The coupling of different loss terms by voltage magnitudes and angles gives rise to a competitive dynamic throughout the training process. To dynamically balance the gradient contributions from different loss terms in (3) in DNN training, we design an adaptive learning rate algorithm in Section IV-B.

As discussed in [6], there could be mismatches between the demanded and satisfied loads due to prediction errors. Unsatisfied loads are also inevitable in conventional methods. Considering power losses in transmission lines, inexact system parameters, etc., a small load-generation imbalance ratio (around 1%) is acceptable [42].

B. Neural Network Training of DeepOPF-NGT

The DNN model of DeepOPF-NGT is trained by minimizing the loss function $\mathcal{L}$ in (3). A simple and viable approach is to apply the gradient descent algorithm. Denote the DNN parameters as ϕ, and the mapping from the input $\boldsymbol{x} = (\boldsymbol{P_d}, \boldsymbol{Q_d})^T$ to the output $\boldsymbol{y} = (\hat{\boldsymbol{V}}_{\boldsymbol{\alpha}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{\alpha}})^T$ as $\boldsymbol{y} = \xi(\boldsymbol{x}, \phi)$. Then, we can obtain $\hat{\boldsymbol{P}}_{\boldsymbol{g}}(\xi(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{Q}}_{\boldsymbol{g}}(\xi(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{P}}_{\boldsymbol{d}}(\xi(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{Q}}_{\boldsymbol{d}}(\xi(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{S}}_{\boldsymbol{l}}(\xi(\boldsymbol{x}, \phi))$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{l}}(\xi(\boldsymbol{x}, \phi))$, where $\hat{\boldsymbol{S}}_{\boldsymbol{l}} = (\hat{S}_{ij}, \forall(i,j) \in \mathcal{E})$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{l}} = (\hat{\theta}_{ij}, \forall(i,j) \in \mathcal{E})$. Besides, denote $\boldsymbol{z} = (\hat{\boldsymbol{V}}_{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{\beta}})^T$. We reformulate the loss function in (3) as below:

$$\mathcal{L} = k_{obj}\mathcal{L}_{obj} + \sum\nolimits_{i=1}^{N_c} k_i\mathcal{L}_i, \tag{10}$$

where $\mathcal{L}_i$ is the loss term for the i-th constraint in (1e)-(1i), k_i is the corresponding coefficient, N_c is the total number

Algorithm 1: Training of DeepOPF-NGT

Input : DNN model with initial parameters $\boldsymbol{\phi}_0$, other initial parameters η_0, k_{obj}, k_g^0, $k_{S_l}^0$, $k_{\theta_l}^0$, k_z^0 and k_d^0, training dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$.

Output : Trained DNN model with parameters $\boldsymbol{\phi}$.

for $t = 1$ **to** T **do**
  Shuffle the training dataset $\mathcal{D}$.
  for *each batch* $\mathcal{B} \subset \mathcal{D}$ **do**
    for $i = 1$ **to** m **do**
      $\boldsymbol{y}_i \leftarrow \hat{\xi}(\boldsymbol{x}_i, \boldsymbol{\phi}_{t-1})$
    end
    Compute zero-injection bus voltages $\boldsymbol{z}$ by (2).
    Compute $\mathcal{L}_{obj}$ and $\{\mathcal{L}_i, i = 1, 2, ..., N_c\}$.
    Update k_i^t by (12) and calculate $\mathcal{L}$.
    $\boldsymbol{\phi}_t \leftarrow \boldsymbol{\phi}_{t-1} - \eta_t \nabla_{\boldsymbol{\phi}_{t-1}} \mathcal{L}$
  end
  $t \leftarrow t + 1$
end

of constraints. The parameters $\boldsymbol{\phi}$ are updated according to $\boldsymbol{\phi}_{t+1} = \boldsymbol{\phi}_t - \eta_t \nabla_{\boldsymbol{\phi}_t} \mathcal{L}$, where η_t is a positive step size at the t-th epoch of training, and the gradient $\nabla_{\boldsymbol{\phi}} \mathcal{L}$ is obtained by using the chain rule as below:

$$\begin{aligned} \nabla_{\boldsymbol{\phi}} \mathcal{L} &= \nabla_{\boldsymbol{y}} \mathcal{L} \cdot \nabla_{\boldsymbol{\phi}} \boldsymbol{y} \\ &= (k_{obj} \nabla_{\boldsymbol{y}} \mathcal{L}_{obj} + \nabla_{\boldsymbol{y}} \sum_{i=1}^{N_c} k_i \mathcal{L}_i) \cdot \nabla_{\boldsymbol{\phi}} \boldsymbol{y} \\ &\quad + (k_{obj} \nabla_{\boldsymbol{z}} \mathcal{L}_{obj} + \nabla_{\boldsymbol{z}} \sum_{i=1}^{N_c} k_i \mathcal{L}_i) \cdot \nabla_{\boldsymbol{y}} \boldsymbol{z} \cdot \nabla_{\boldsymbol{\phi}} \boldsymbol{y}. \end{aligned} \tag{11}$$

All terms in $\nabla_{\boldsymbol{y}} \mathcal{L}$ can be derived from (1)-(9). For faster convergence, we employ the mini-batch stochastic gradient descent algorithm to update the DNN parameters based on a subset of the training data rather than the entire dataset [43].

The key challenge of training DeepOPF-NGT lies in gradient pathologies mainly caused by the gradient imbalance between different loss terms in (3), a common issue when applying gradient decent method [37]. The dominant loss term may bias the DNN training towards neglecting the contribution of others. To address this issue, we develop an adaptive learning rate algorithm based on the learning rate annealing algorithm developed in [37]. Different from [37], we directly balance different loss terms and add upper bounds (fine-tuned manually) to their coefficients to prevent gradient explosion, which helps jump out of local optima and speed up convergence for DNN training. For each training epoch $t > 1$, given a fixed coefficient k_{obj}, the coefficient k_i^t is updated as follows:

$$k_i^t = \min(k_{obj} \mathcal{L}_{obj} / \mathcal{L}_i, \overline{k}_i), \tag{12}$$

where $\overline{k}_i$ is the upper bounds of k_i^t. The coefficient k_i^t is calculated per constraint by summing up the terms $\mathcal{L}_{obj}$ and $\mathcal{L}_i$ over all training samples in each mini-batch. Then, we developed Algorithm 1 for the training of DeepOPF-NGT. Fig. 4 presents the flowchart of the proposed DeepOPF-NGT.

C. Discussion

This study employs the widely used mini-batch gradient descent algorithm to search for optimal parameters of DNNs.

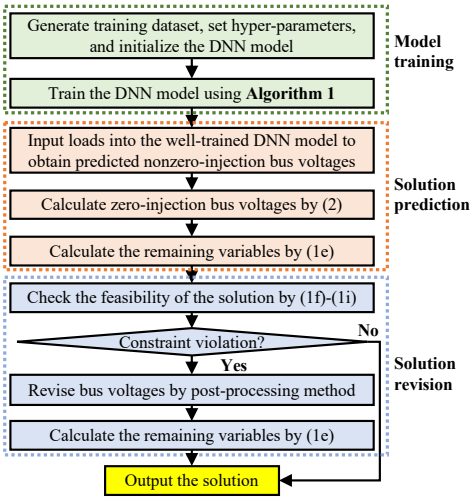


Fig. 4. Flowchart illustrating the training and application of DeepOPF-NGT.

Several common issues of DNN training may exist. First, although this algorithm can escape local minima to reach a good minimum, it is unable to guarantee a global optimum. As a result, the obtained parameters of DNNs may be suboptimal, and different mappings may be obtained in different runs. Hence, the predicted AC-OPF solutions may be suboptimal. However, existing AC-OPF solvers can only provide suboptimal solutions as well. To date, finding globally optimal solutions for AC-OPF problems under general settings remains an open problem. To avoid getting stuck in local optima, we develop an adaptive learning algorithm. Second, the sharp minima issue may prevent DNNs from generalizing well with the testing dataset. Several metaheuristic algorithms have been proposed to alleviate this issue and can be applied to improve the generalization capability of DeepOPF-NGT [44]. Despite these limitations that exist for most machine learning approaches, DeepOPF-NGT does not require ground truths compared with existing supervised learning methods and has the potential to find better solutions than conventional solvers.

Regarding the scalability of DeepOPF-NGT, on one hand, since there are a considerable number of ZIBs in power systems, using Kron reduction can reduce the number of prediction variables significantly. For instance, the IEEE 300-bus system has 67 ZIBs. On the other hand, the scalability can be further improved by developing a distributed training algorithm, which will be our future work. It involves partitioning the predicted variables into multiple groups and employing a deep neural network model to predict each group. During the training process, models associated with interdependent variables will engage in the exchange of predicted variables, enabling the calculation of their respective loss functions. This approach holds the potential to substantially reduce the neural network size, resulting in expedited training times.

V. Performance Analysis and Extension of DeepOPF-NGT

A. Learning a Legitimate Mapping

In this subsection, we explore sufficient conditions for learning a legitimate mapping (see Definiton 3) by the proposed

unsupervised learning approach DeepOPF-NGT. We start by formalizing the notions. We consider a standard AC-OPF problem and assume there are M continuous target load-solution mappings [5]. Each target mapping is denoted as $\xi_i : \boldsymbol{x} \to \boldsymbol{y}, i = 1, 2, ..., M$, where $\boldsymbol{x} \in \mathbb{R}^{2N_L}$ and $\boldsymbol{y} \in \mathbb{R}^{2(N-N_Z)}$ are the input (loads) and the output (ground truths of nonzero-injection bus voltages), respectively, and N, N_L and N_Z are cardinalities of $\mathcal{N}$, $\mathcal{N}_L$ and $\mathcal{N}_Z$, respectively. The set of all target mappings is denoted as $\boldsymbol{\xi}$. For instance, target mapping 1 (denoted as ξ_1) and target mapping 2 (denoted as ξ_2) in Fig. 2 form the set $\boldsymbol{\xi} = \{\xi_1, \xi_2\}$. The mapping from $\boldsymbol{y}$ to the loss function $\mathcal{L}$ is represented by $\psi : \boldsymbol{y} \to \mathcal{L}$. It is known from (3) that $\psi(\boldsymbol{y})$ is a scalar. The feasible sets (in the optimization sense) of $\boldsymbol{x}$, $\boldsymbol{y}$ are denoted as $\mathcal{X}$ and $\mathcal{Y}$, respectively. The mapping learned by DNNs is denoted as ξ_{nn}. In the compact set $\mathcal{X}$, ξ_i is Lipschitz continuous except for a set of Lebesgue measure zero [5], and ξ_{nn} is Lipschitz continuous over $\mathcal{X}$ [16].

Regarding ψ, we make the following assumption:

- A1: $\psi(\boldsymbol{y})$ is Lipschitz continuous, i.e., there exists a constant $L_\psi > 0$ such that $|\psi(\boldsymbol{y}_i) - \psi(\boldsymbol{y}_j)| \le L_\psi \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$ for any $\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathcal{Y}$.

Assumption A1 requires ψ to be smooth. The rationale behind A1 is that the power generation and load change continuously with bus voltage magnitudes and angles according to equations (1b)-(1e). As a result, the loss function $\mathcal{L}$ also changes continuously with bus voltages $\boldsymbol{y}$.

Before proceeding, we provide the following definitions for our analysis. First, we define whether two target mappings ξ_i and ξ_j can be distinguished based on their distances.

Definition 1. *For a given $\varepsilon > 0$ and a compact set $\mathcal{X}$, we say two mappings ξ_i and ξ_j are ε-similar to each other in $\mathcal{X}$ if $\|\xi_i(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \le \varepsilon$ for all $\boldsymbol{x} \in \mathcal{X}$.*

Note that "ε-similar" means "not ε-distinguishable". In AC-OPF problems, we observed that in some subsets of $\mathcal{X}$, there exists $\varepsilon > 0$ such that ξ_i and ξ_j are ε-distinguishable, while in other subsets of $\mathcal{X}$, we can not find $\varepsilon > 0$ such that ξ_i and ξ_j can be distinguished. Thus, we further define the ε-similar set.

Definition 2. *For a given $\varepsilon > 0$ and two mappings $\xi_i, \xi_j \in \boldsymbol{\xi}$ in a compact set $\mathcal{X}$, the ε-similar set $\mathcal{X}_{ij}^{\varepsilon} \subseteq \mathcal{X}$ is the set where ξ_i and ξ_j are ε-similar to each other.*

Based on the above, we define the legitimate mapping as follows.

Definition 3. *For a given $\delta > 0$ and a compact set $\mathcal{X}$, ξ_{nn} is a legitimate mapping, if there exists one and only one target mapping $\xi_i \in \boldsymbol{\xi}$ such that $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_i(\boldsymbol{x}))| \le \delta$ for all $\boldsymbol{x} \in \mathcal{X}$.*

Fig. 5 presents an example of the legitimate mapping, where ξ_1 (marked by blue curve) and ξ_2 (marked by orange curve) are two different target mappings. In the orange dash area, $|\psi(\xi_{nn,1}(\boldsymbol{x})) - \psi(\xi_1(\boldsymbol{x}))| \le \delta$; In the blue dash area, $|\psi(\xi_{nn,2}(\boldsymbol{x})) - \psi(\xi_2(\boldsymbol{x}))| \le \delta$. By Definition 3, $\xi_{nn,1}$ (marked by green curve) and $\xi_{nn,2}$ (marked by red curve) are legitimate mappings, while $\xi_{nn,3}$ (marked by brown curve) and $\xi_{nn,4}$ (marked by purple curve) are not legitimate mappings.

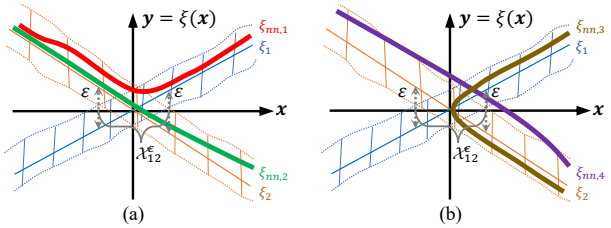


Fig. 5. An example of a single-input single-output mapping.

Then, we derive sufficient conditions for DNNs to learn a legitimate mapping in Theorem 1.

Theorem 1. *Suppose A1 holds and $\mathcal{X}$ is compact. ξ_{nn} is a legitimate mapping if there exists $\varepsilon > 0$ and $\delta > 0$ such that C1–C3 are satisfied:*

C1: $\forall \boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_{ij}^{\varepsilon}$, $\forall \xi_i, \xi_j \in \boldsymbol{\xi}$, $\|\xi_i(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \ge \varepsilon$.

C2: $\forall \boldsymbol{x} \in \mathcal{X}$, $\exists \xi_i \in \boldsymbol{\xi}$, $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_i(\boldsymbol{x}))| \le \delta$.

C3: $\delta < \frac{\varepsilon L_\psi}{2}$.

Proof: See Appendix A. ∎

In practice, for $\boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_{ij}^{\varepsilon}$, C1 will be satisfied if the target mappings are distinguishable, i.e., the deviation between the zero-order derivatives of any two target mappings $\xi_i, \xi_j \in \boldsymbol{\xi}$ is sufficiently large; for $\boldsymbol{x} \in \mathcal{X}_{ij}^{\varepsilon}$, ξ_i and ξ_j are too close to each other to be distinguished. In C2, $\psi(\xi_i(\boldsymbol{x}))$ only consists of the objective, whereas $\psi(\xi_{nn}(\boldsymbol{x}))$ consists of the objective and penalties for constraint violations and unsatisfied loads. Thus, C2 will be satisfied when the DNNs achieve good optimality and feasibility performances. C3 implies that it is easier to learn a legitimate mapping when target mappings are sufficiently different from each other and the loss function is small.

To our knowledge, this is the first theoretical justification of learning a legitimate mapping for non-convex optimization problems (not restricted to OPF problems) by *unsupervised* learning. It demonstrates the potential of the proposed *unsupervised* learning approach in addressing the fundamental multiple mapping issues in the supervised learning approach.

B. Extension to Semi-Supervised Learning Approach

Now we consider the situation where there are a few samples with ground truths. Since the ground truth contains the information of the test system, it can be leveraged to pre-train the DNN model instead of learning from scratch. In the pre-training process, the supervised learning approach is applied with the loss function as below:

$$\mathcal{L}' = k_v \mathcal{L}_v + \sum\nolimits_{i=1}^{N_c} k_i \mathcal{L}_i, \tag{13}$$

where k_v is a positive constant, and $\mathcal{L}_v$ is the prediction error as below:

$$\mathcal{L}_v = \sum\nolimits_{i \in \mathcal{N}} \left(||\hat{V}_i - V_i||_2^2 + ||\hat{\theta}_i - \theta_i||_2^2 \right). \tag{14}$$

In this way, DeepOPF-NGT is extended to a semi-supervised learning approach.

To fully utilize the ground truths, we propose Algorithm 2 to pre-train the DNN model at each training epoch, where $\mathcal{D}$ and $\mathcal{D}'$ are datasets with and without ground truths, respectively, and $\nabla_{\boldsymbol{\phi}} \mathcal{L}'$ is derived using the chain rule. In the ideal case,

Algorithm 2: Training of extended DeepOPF-NGT

Input : DNN model with initial parameters $\boldsymbol{\phi}_0$, other initial parameters η_0, k_v, k_g^0, $k_{S_l}^0$, $k_{\theta_l}^0$, k_z^0, and k_d^0, training datasets $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ and $\mathcal{D}' = \{(\boldsymbol{x}_{n+1}, \boldsymbol{y}_{n+1}), ..., (\boldsymbol{x}_{n+n'}, \boldsymbol{y}_{n+n'})\}$.

Output : Trained DNN model with parameters $\boldsymbol{\phi}$.

for $t = 1$ **to** T **do**
- Shuffle the training datasets $\mathcal{D}$ and $\mathcal{D}'$.
- *Pre-training process with ground truths*:
- **for** *each batch* $\mathcal{B}' \subset \mathcal{D}'$ **do**
  - **for** $i = 1$ **to** m **do**
    - $\boldsymbol{y}_i \leftarrow \hat{\xi}(\boldsymbol{x}_i, \boldsymbol{\phi}_{t-1})$
  - **end**
  - Compute zero-injection bus voltages $\boldsymbol{z}$ by (2).
  - Compute $\mathcal{L}_v$ and $\{\mathcal{L}_i, i = 1, 2, ..., N_c\}$.
  - Update k_i^t by (12) and calculate $\mathcal{L}'$.
  - $\boldsymbol{\phi}_t' \leftarrow \boldsymbol{\phi}_{t-1} - \eta_t \nabla_{\boldsymbol{\phi}_{t-1}} \mathcal{L}'$
- **end**
- *Training process without ground truths*:
- **for** *each batch* $\mathcal{B} \subset \mathcal{D}$ **do**
  - **for** $i = 1$ **to** m **do**
    - $\boldsymbol{y}_i \leftarrow \hat{\xi}(\boldsymbol{x}_i, \boldsymbol{\phi}_t')$
  - **end**
  - Compute zero-injection bus voltages $\boldsymbol{z}$ by (2).
  - Compute $\mathcal{L}_{obj}$ and $\{\mathcal{L}_i, i = 1, 2, ..., N_c\}$.
  - Update k_i^t by (12) and calculate $\mathcal{L}$.
  - $\boldsymbol{\phi}_t \leftarrow \boldsymbol{\phi}_t' - \eta_t \nabla_{\boldsymbol{\phi}_t'} \mathcal{L}$
- **end**
- $t \leftarrow t + 1$

end

the ground truths should come from the same load-solution mapping. One possible method is to use a fixed initial point when generating ground truths using a conventional OPF solver[3]. Nevertheless, when the number of samples with ground truths is substantially smaller than that without ground truths, there may be little influence even if the ground truths are from different mappings.

C. Discussion

Distinguishing between different load-solution mappings remains a challenging open problem. Thus, it is also hard to determine whether the learned mapping is a legitimate mapping when the conditions in Theorem 1 are not satisfied. These intriguing questions provide directions for future investigation beyond the scope of this study. In this work, we apply common metrics to evaluate the feasibility and optimality of DeepOPF-NGT (see Section VI-A). While DeepOPF-NGT may not guarantee learning a legitimate mapping due to imperfect training, the well-trained model will still achieve good performances as long as the training loss is small, as verified in Section VI.

[3] It does not guarantee that the samples are from the same mapping but aims to avoid generating samples from multiple mappings. Given a fixed initial point, the IPOPT solver will converge to a deterministic solution for each input [45].

VI. Numerical Experiments

This section presents the numerical tests conducted on the modified IEEE 39/118/300-bus systems and the modified 1354-bus system to validate the proposed DeepOPF-NGT. Section VI-A illustrates the experimental setup. Section VI-B compares DeepOPF-NGT with state-of-the-art supervised learning approaches. Section VI-C evaluates DeepOPF-NGT against both supervised and unsupervised learning approaches to assess its capability in addressing multiple mapping issues. Section VI-D validates the scalability of DeepOPF-NGT. Section VI-E verifies the efficiency of the proposed adaptive learning rate algorithm. Section VI-F validates the effectiveness of the extended DeepOPF-NGT.

A. Experimental Setup

1) Comparing methods: The following typical state-of-the-art machine learning methods are considered for comparison:

- **DeepOPF-V** [6]: It learns the mapping between loads and voltages of all buses. For a fair comparison, we add penalties for constraint violations and load deviations to the loss function in [6], i.e., using $\mathcal{L}'$ in (13).
- **DeepOPF-AC** [5]: It learns the mapping between loads and the optimal generation setpoints (i.e., active power generation and voltage magnitude of generation buses) and reconstructs the remaining variables by solving power balance equations.
- **EACOPF** [46]: It predicts the generation setpoints using a well-trained model that emulates an iterative solver and reconstructs the remaining variables by solving power balance equations. And it requires a large training dataset.
- **DC3** [32]: It is an unsupervised learning approach. It first predicts the optimal generation setpoints, then calculates the remaining bus voltages using Newton's method, and finally obtains the remaining variables via power balance equations.

2) Performance evaluation: It is hard to visualize the high-dimensional mapping learned by DNNs. To this end, we regard the solution provided by the IPOPT solver as the benchmark and apply the following metrics to evaluate the performance:

- ***Speedup Factor***: It is the average ratio of the computation time consumed by the IPOPT solver to solve the original AC-OPF problem to the computation time t_{dnn} consumed by the DNN-based method. It is denoted as η_{sp}.
- ***Optimality Loss***: It measures the average relative deviation between the optimal objective found by the IPOPT solver and that found by the DNN-based method.
- ***Constraint Satisfaction Ratio***: It evaluates the feasibility of the predicted solutions by the percentage of satisfied bound constraints. The constraint satisfaction ratios of active power generation, reactive power generation, bus voltage, branch power flow and branch angle are denoted by $\eta_{\boldsymbol{P_g}}$, $\eta_{\boldsymbol{Q_g}}$, $\eta_{\boldsymbol{V}}$, $\eta_{\boldsymbol{S_l}}$ and $\eta_{\boldsymbol{\theta_l}}$, respectively.
- ***Load Satisfaction Ratio***: It is defined as the percentage of loads satisfied for the whole system. The active and reactive load satisfaction ratios are denoted by $\eta_{\boldsymbol{P}_d}$ and $\eta_{\boldsymbol{Q}_d}$, respectively.

TABLE II
GENERAL PARAMETER SETTINGS FOR DNN MODELS.

System	Approach	Batch size	Learning rate	Epoch	Hidden layers
39-bus	DeepOPF-NGT	50	1e-3	2500	256-256
	DeepOPF-V	50	1e-3	12000	256-256
	DC3	200	1e-3	1000	200-200
118-bus	DeepOPF-NGT	50	1e-3	3000	512-256
	DeepOPF-V	50	1e-4	6000	512-256
	DeepOPF-AC	32	1e-3	1000	256-128
	EACOPF	200	1e-6	4000	800
300-bus	DeepOPF-NGT	50	1e-4	2500	512-512
	DeepOPF-V	50	1e-3	6000	1024-768
	DeepOPF-AC	32	1e-3	1000	512-256
	EACOPF	200	1e-6	4000	800

3) Dataset generation: The training dataset contains only a set of load scenarios for the unsupervised learning scheme DeepOPF-NGT and DC3. For the benchmarking supervised learning approaches, the training dataset consists of load scenarios along with their corresponding ground truths which are optimal solutions for AC-OPF problems generated by the IPOPT solver. Note that alternative solvers can also be employed. To investigate the impact of the mixed samples from multiple mappings on the supervised learning approaches, we generated two types of datasets. The first type was generated by using the same default initial point for all load scenarios (in the IEEE 118/300-bus systems), whereas the second type was generated by using different initial points to find multiple global/local solutions (in the IEEE 39-bus system) [41].

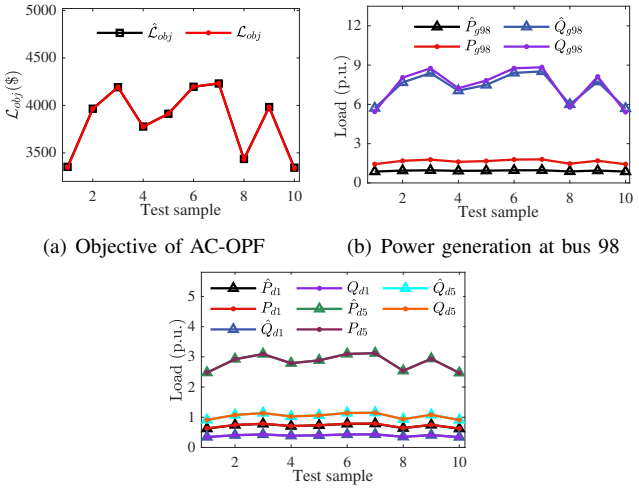
For the 118/300-bus systems, there are 600 training samples for DeepOPF-NGT, 2000 training samples for DeepOPF-V and DeepOPF-AC, and 100,000 training samples for EACOPF. All methods have 2500 test samples. The dataset for the IEEE 39-bus system contains 2000 training samples and 500 test samples. The second type of dataset for DeepOPF-NGT includes a higher number of training samples due to more binding constraints. See [41] for more details on finding multiple local or global optimal solutions. All load scenarios are generated by scaling the default load using the normalized daily total load profile of Bonneville Power Administration on 02/08/2016 from 06:00 am to 12:00 pm [47], enabling us to cover a wide range of load variations up to 42.3%. For the modified 1354-bus system [48], the dataset has 600 training samples and 200 test samples, with 5% load variations.

4) Hyperparameters of DNNs: The DNN-based models are implemented using the PyTorch platform. The hyperparameters are fine-tuned through trial and error (refer to Table II). Other crucial hyperparameters are set as follows: $k_v = 100$, $k_g^0 = k_{S_l}^0 = k_{\theta_l}^0 = k_d^0 = k_z^0 = 1$. Most parameters are configured by [5], [6], [37], [46] and fine-tuned as best as we could.

5) Runtime environment: The models are all trained on a single GPU, while the experimental tests are run on a 64-bit MacBook with 8-core CPU and 32GB RAM. To ensure the reliability of the results, each experimental test was repeated three times to calculate the average performance.

B. Comparison with Supervised Learning Approaches

DeepOPF-NGT is compared with three state-of-the-art supervised learning approaches, i.e., DeepOPF-V [6], DeepOPF-AC [5] and EACOPF [46], in the modified IEEE 118-bus and 300-bus systems. DC3 is not compared due to numerical issues caused by the inverse of the Jacobian matrix during training. Note that the load variation is up to 42.3%, which is larger than that (up to 10%) in [6]. Table III shows that DeepOPF-NGT has comparable performance with the supervised learning methods, but does not require ground truths for training.



(a) Objective of AC-OPF (b) Power generation at bus 98

(c) Loads at buses 1 and 5

Fig. 6. Results of DeepOPF-NGT in the modified IEEE 300-bus system.

Table III reveals that DeepOPF-NGT and DeepOPF-V outperform DeepOPF-AC and EACOPF in terms of computational speed, with speedups of up to three orders of magnitude compared to the IPOPT solver. That is because DeepOPF-AC and EACOPF still need to solve the non-linear power balance equations, while DeepOPF-NG and DeepOPF-V only require simple matrix operations. Additionally, the optimality loss of DeepOPF-NGT and DeepOPF-V are both less than 1.0%, while that of DeepOPF-AC and EACOPF can be up to 3.6% and 10.7% in the 118-bus system, respectively.

Concerning the feasibility of the solution, most inequality constraints are satisfied in these four approaches. EACOPF has the smallest $\eta_{\boldsymbol{P}_g}$ that can be smaller than 98% in the 300-bus system. Meanwhile, DeepOPF-NGT and DeepOPF-V can ensure the satisfaction of voltage constraints, while DeepOPF-AC and EACOPF cannot. The reason is that DeepOPF-NGT and DeepOPF-V obtain bus voltages directly and thus can keep them within limits, while DeepOPF-AC and EACOPF reconstruct bus voltages by solving power flow equations using predicted generation setpoints, and thus the voltage constraints may be violated. Thus, DeepOPF-NGT and DeepOPF-V always guarantee to obtain a solution since power flow equations are satisfied automatically, while DeepOPF-AC and EACOPF may have no solution due to prediction errors. Both DeepOPF-NGT and DeepOPF-V exhibit acceptable load satisfaction ratios, exceeding 99%. There is a slight violation of the branch flow constraint in DeepOPF-NGT due to the binding of one branch flow constraint.

Fig. 6 shows the comparison results of the predicted solutions and their ground truths for DeepOPF-NGT in the 300-bus system. Due to space limitations, we only present 10 samples selected uniformly at random in the test dataset. It shows that the predicted objective is close to its ground truth, despite the

TABLE III
COMPARISON RESULTS IN THE MODIFIED IEEE 118-BUS AND 300-BUS SYSTEMS.

Metric	IEEE 118-bus system				IEEE 300-bus system			
	DeepOPF-NGT	DeepOPF-V	DeepOPF-AC	EACOPF	DeepOPF-NGT	DeepOPF-V	DeepOPF-AC	EACOPF
$\eta_{opt}(\%)$	<1.0	<0.1	<3.6	<10.7	<0.1	<1.0	<0.6	<0.8
$\eta_{\boldsymbol{V}}(\%)$	-	-	**99.2**	**94.5**	-	-	**99.2**	**90.8**
$\eta_{\boldsymbol{P_g}}(\%)$	100.0	100.0	100.0	99.3	100.0	100.0	100.0	**97.9**
$\eta_{\boldsymbol{Q_g}}(\%)$	100.0	99.9	99.9	100.0	100.0	99.3	100.0	100.0
$\eta_{\boldsymbol{S_l}}(\%)$	99.6	100.0	100.0	99.5	99.9	100.0	100.0	99.9
$\eta_{\boldsymbol{\theta_l}}(\%)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$\eta_{\boldsymbol{P_d}}(\%)$	99.8	99.9	-	-	99.8	**92.3**	-	-
$\eta_{\boldsymbol{Q_d}}(\%)$	99.9	100.0	-	-	99.8	**93.7**	-	-
$t_{dnn}(s)$	1.9e-4	7.2e-4	1.3e-2	3.2e-2	2.4e-4	3.2e-4	2.0e-2	6.2e-2
η_{sp}	×3589	×1995	×50	×21	×7588	×5731	×95	×30

TABLE IV
RESULTS OF DEEPOPF-NGT IN THE MODIFIED 1354-BUS SYSTEM.

η_{opt}	$\eta_{\boldsymbol{V}}$	$\eta_{\boldsymbol{P_g}}$	$\eta_{\boldsymbol{Q_g}}$	$\eta_{\boldsymbol{S_l}}$	$\eta_{\boldsymbol{\theta_l}}$	$\eta_{\boldsymbol{P_d}}$	$\eta_{\boldsymbol{Q_d}}$	t_{dnn}	η_{sp}	Batch size	Learning rate	Epoch	Hidden layers
<1%	-	>99%	>99%	>99%	100%	99%	99%	4.8e-3s	×841	50	1e-4	3000	3072-3072

small deviations between the predicted power generations and their ground truths in Fig. 6(b). Fig. 6(c) shows the predicted and actual loads are nearly identical.

C. Performance in Handling Multiple Mappings

To verify the effectiveness of DeepOPF-NGT in addressing the issue of multiple load-solution mappings, we compare it with the supervised learning approach DeepOPF-V and the unsupervised learning approach DC3 in the modified IEEE 39-bus system. DeepOPF-NGT and DeepOPF-V have the same input (loads) and similar output (bus voltage magnitudes and angles) but different loss functions. The dataset consists of samples from multiple load-solution mappings with an average of 30% difference in objectives. The numbers of the low-cost and high-cost solutions are equal for each load input.

Table V indicates that both DeepOPF-NGT and DeepOPF-V exhibit good performance in feasibility (i.e., all constraints are satisfied) and computational speedup (i.e., up to three orders of magnitude). Although DeepOPF-NGT has a little larger optimality loss than DeepOPF-V, it surpasses DeepOPF-V in terms of reactive load satisfaction ratio. As for DC3, it satisfies all loads but violates voltage and reactive power generation constraints. Besides, it has a much smaller computational speedup than DeepOPF-NGT and DeepOPF-V. Thus, DeepOPF-NGT achieves the best performance when there are multiple load-solution mappings embedded in the dataset.

D. Scalability of the Proposed DeepOPF-NGT

To validate the scalability of our proposed DeepOPF-NGT, we conducted numerical tests in the modified 1354-bus systems. As shown in Table IV, the optimality loss is less than 1%, the constraint and load satisfaction ratios are all over 99%, and the computational speedup is over ×800. These results indicate that DeepOPF-NGT continues to exhibit good performance even when applied to a larger system, and thus has good scalability.

TABLE V
COMPARISON RESULTS IN THE MODIFIED IEEE 39-BUS SYSTEM.

Metric	DeepOPF-NGT	DeepOPF-V	DC3
$\eta_{opt}(\%)$	<5	<1	<10
$\eta_{\boldsymbol{V}}(\%)$	-	-	**95**
$\eta_{\boldsymbol{P_g}}(\%)$	100	100	100
$\eta_{\boldsymbol{Q_g}}(\%)$	100	100	**83**
$\eta_{\boldsymbol{S_l}}(\%)$	100	100	99
$\eta_{\boldsymbol{\theta_l}}(\%)$	100	100	100
$\eta_{\boldsymbol{P_d}}(\%)$	>99	>99	-
$\eta_{\boldsymbol{Q_d}}(\%)$	>99	< **89**	-
t_{dnn}	1.4e-4	1.2e-4	2.4e-3
η_{sp}	×5415	×6643	×**323**

TABLE VI
COMPARISON RESULTS OF DIFFERENT LEARNING RATE ALGORITHMS IN THE MODIFIED IEEE 118-BUS SYSTEM.

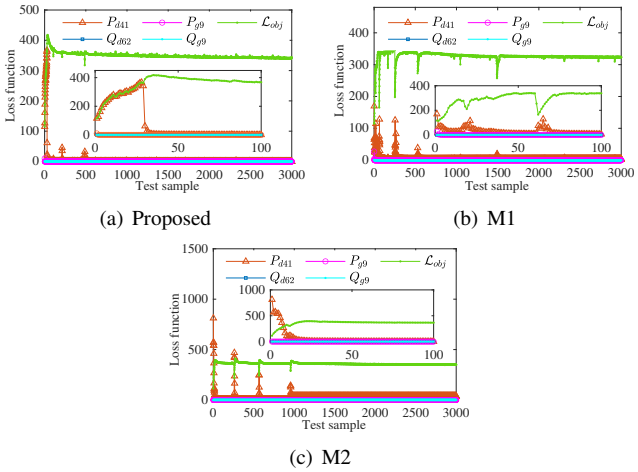
Method	Proposed	M1	M2
$\eta_{opt}(\%)$	< 1.0	< 3.3	−5.0
$\eta_{\boldsymbol{V}}(\%)$	-	-	-
$\eta_{\boldsymbol{P_g}}(\%)$	100.0	**98.7**	**98.7**
$\eta_{\boldsymbol{Q_g}}(\%)$	100.0	100.0	100.0
$\eta_{\boldsymbol{S_l}}(\%)$	99.6	99.5	99.5
$\eta_{\boldsymbol{\theta_l}}(\%)$	100	100.0	100.0
$\eta_{\boldsymbol{P_d}}(\%)$	99.8	**92.6**	**92.7**
$\eta_{\boldsymbol{Q_d}}(\%)$	99.9	**96.7**	**96.6**

E. Performance of Adaptive Learning Rate Algorithm

We compare the proposed adaptive learning rate algorithm with the following two methods in the modified IEEE 118-bus system to demonstrate its effectiveness:

- M1: Fixed coefficients for different terms in the loss function, a common method in existing studies;
- M2: Learning rate annealing algorithm [37].

Table VI indicates that our proposed method outperforms the other two methods, achieving less than 1.0% optimality loss and more than 99% load satisfaction ratios. In contrast, M1 and M2 exhibit low load satisfaction ratios, less than 95%. Since



(a) Proposed (b) M1

(c) M2

Fig. 7. Different terms of the loss function during training in the modified IEEE 118-bus system.

TABLE VII
EXTENDED DEEPOPF-NGT IN THE MODIFIED IEEE 300-BUS SYSTEM.

N_{label}	0	50	150	250
$\eta_{opt}(\%)$	< 0.3	< 0.2	< 0.1	< 0.1
$\eta_{\boldsymbol{V}}(\%)$	-	-	-	-
$\eta_{\boldsymbol{P_g}}(\%)$	**99.1**	99.4	99.7	**99.8**
$\eta_{\boldsymbol{Q_g}}(\%)$	99.9	100.0	100.0	100.0
$\eta_{\boldsymbol{S_l}}(\%)$	100.0	100.0	100.0	100.0
$\eta_{\boldsymbol{\theta_l}}(\%)$	100.0	100.0	100.0	100.0
$\eta_{\boldsymbol{P_d}}(\%)$	**99.4**	99.5	99.6	**99.8**
$\eta_{\boldsymbol{Q_d}}(\%)$	**98.8**	99.2	99.4	**99.7**

the loss function is updated dynamically during training, using fixed coefficients in M1 may lead to dominant loss terms, which can cause unbalanced back-propagated gradients and bias the training towards ignoring the contributions of small loss terms [37]. Regarding M2, it may have exploding gradients due to unbounded coefficients for different loss terms. As shown in Fig. 7, our proposed method has more balanced loss terms than the other methods at the early stage of training. As the training goes on, only when the constraint violations and load deviations are minor and the coefficients k_g, k_{S_l},k_{θ_l} reach their upper bounds, there will be unbalanced loss terms. This result indicates that it is important to balance different loss terms at the early stage of training.

F. Performance of Extended DeepOPF-NGT

To explore how ground truths can improve the performance of DeepOPF-NGT, we compare DeepOPF-NGT with its extended version in the modified IEEE 300-bus system. We only use 300 training samples without ground truths and N_{label} samples with ground truths generated by the IPOPT solver using a fixed initial point. When the number of training samples decreases from 600 to 300, the performance of DeepOPF-NGT becomes worse. The results in Table III and Table VII show that $\eta_{\boldsymbol{P_g}}$ and $\eta_{\boldsymbol{Q_d}}$ decrease from 100.0% and 99.9% to 99.1% and 98.8%, respectively. However, all performances tend to be improved as N_{label} increases (see Table VII), showing an obvious tendency that the ground truth can effectively improve DeepOPF-NGT. Notably, the optimality loss is negative, which can be attributed to the fact that some loads are not fully satisfied, resulting in less power generation. Additionally, DeepOPF-NGT attempts to find the solution with the smallest objective during the training, so it may find a better solution than the IPOPT solver which may give a locally optimal solution.

VII. CONCLUSION

We propose an *unsupervised* learning approach called DeepOPF-NGT for solving AC-OPF problems efficiently without ground truths. It directly learns a legitimate load-solution mapping under the guidance of a properly designed loss function consisting of the objective and penalties for constraint violations and dissatisfied loads of the AC-OPF problem. Kron reduction is used to improve the scalability and feasibility. An adaptive learning rate algorithm is devised to handle unbalanced gradient pathologies. We have also derived the first condition (to our best knowledge) for *unsupervised* learning to learn a legitimate mapping. The results on the modified 118-bus and 300-bus systems verify that DeepOPF-NGT has comparable performance with the state-of-the-art supervised learning approaches. It provides feasible solutions with minor optimality loss (less than 1.0%) and a decent computational speedup (up to three orders of magnitude faster than the IPOPT solver). While in the modified IEEE 39-bus system with multiple load-solution mappings embedded in the dataset, DeepOPF-NGT achieves better performance than existing supervised and unsupervised learning methods. The results in the modified 1354-bus system further validate its scalability. The effectiveness of the adaptive learning rate method has also been verified. Moreover, the results indicate that a few ground truths can help enhance DeepOPF-NGT.

An interesting future direction is to design a neural network architecture to balance the interplay between different terms in the loss function and explore more efficient training algorithms.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for constructive comments that further help improve the work.

APPENDIX

A. Proof of Theorem 1

Assume ξ_{nn} is not a legitimate mapping. We will show by contradiction that this assumption violates C1–C3 in Theorem 1. For $\boldsymbol{x} \in \mathcal{X}_{ij}^{\varepsilon}$, we do not differentiate the target mappings because they are too close to each other. Condition C2 ensures that ξ_{nn} has good optimality and feasibility performances. Thus, we will focus on the remaining region $\boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_{ij}^{\varepsilon}$.

Assume there exists $\boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_{ij}^{\varepsilon}$ such that ξ_{nn} consists of two target mappings ξ_i, $\xi_j \in \boldsymbol{\xi}$. By assumption A1, $\forall \boldsymbol{x} \in \mathcal{X}$, $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_i(\boldsymbol{x}))| \leq L_\psi \|\xi_{nn}(\boldsymbol{x}) - \xi_i(\boldsymbol{x})\|_2$, and $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_j(\boldsymbol{x}))| \leq L_\psi \|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2$. Condition C2 requires that $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_i(\boldsymbol{x}))| \leq \delta$ and $|\psi(\xi_{nn}(\boldsymbol{x})) - \psi(\xi_j(\boldsymbol{x}))| \leq \delta$. To ensure that C2 is satisfied, we should have $L_\psi \|\xi_{nn}(\boldsymbol{x}) - \xi_i(\boldsymbol{x})\|_2 \leq \delta$, and $L_\psi \|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \leq \delta$, resulting in $\|\xi_{nn}(\boldsymbol{x}) - \xi_i(\boldsymbol{x})\|_2 \leq \delta / L_\psi$ and $\|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \leq \delta / L_\psi$. Then, we have

$$\|\xi_{nn}(\boldsymbol{x}) - \xi_i(\boldsymbol{x})\|_2 + \|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \leq 2\delta / L_\psi \quad (15)$$

Since $\|\xi_i(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 = \|\xi_i(\boldsymbol{x}) - \xi_{nn}(\boldsymbol{x}) + \xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \leq \|\xi_i(\boldsymbol{x}) - \xi_{nn}(\boldsymbol{x})\|_2 + \|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2$, we have the following inequation by C1:

$$\|\xi_i(\boldsymbol{x}) - \xi_{nn}(\boldsymbol{x})\|_2 + \|\xi_{nn}(\boldsymbol{x}) - \xi_j(\boldsymbol{x})\|_2 \geq \varepsilon \tag{16}$$

According to (15)–(16), we obtain $\delta \geq \varepsilon L_\psi/2$ which contradicts C3. Thus, the assumption does not hold, and ξ_{nn} is a legitimate mapping.

REFERENCES

[1] A. Schecter and R. P. O'Neill, "Exploration of the ACOPF feasible region for the standard IEEE test set," *FERC Staff Technical Paper*, 2013.

[2] F. Capitanescu, "Critical review of recent advances and further developments needed in AC optimal power flow," *Electr. Power Syst. Res.*, vol. 136, pp. 57–68, 2016.

[3] M. Niu, C. Wan, and Z. Xu, "A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems," *J. Mod. Power Syst. Clean Energy*, vol. 2, no. 4, pp. 289–297, 2014.

[4] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems-A review," *IEEE Trans. Power Syst.*, 2022.

[5] X. Pan, W. Huang, M. Chen, and S. H. Low, "DeepOPF-AL: augmented learning for solving AC-OPF problems with a multi-valued load-solution mapping," in *ACM e-Energy*, 2023, pp. 42–47.

[6] W. Huang, X. Pan, M. Chen, and S. H. Low, "DeepOPF-V: Solving AC-OPF Problems Efficiently," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 800–803, 2022.

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

[8] S. Pineda, J. M. Morales, and A. Jiménez-Cordero, "Data-driven screening of network constraints for unit commitment," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3695–3705, 2020.

[9] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for DCOPF with generalization guarantees," *IEEE Trans. Control. Netw. Syst.*, vol. 9, no. 2, pp. 719–730, 2021.

[10] Y. Ng, S. Misra, L. A. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *Proc. Power Syst. Comput. Conf.,*, Dublin, Ireland, 2018, pp. 1–7.

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

[12] D. Biagioni, P. Graf, X. Zhang, A. S. Zamzam, K. Baker, and J. King, "Learning-accelerated ADMM for distributed DC optimal power flow," *IEEE Control Syst. Lett.*, vol. 6, pp. 1–6, 2020.

[13] F. Diehl, "Warm-starting AC optimal power flow with graph neural networks," in *Proc. 33rd Conf. Neural Inf. Process. Syst.,*, 2019, pp. 1–6.

[14] W. Dong, Z. Xie, G. Kestor, and D. Li, "Smart-PGSim: Using neural network to accelerate AC-OPF power grid simulation," in *Proc SC20: Int. Conf. High Perform. Comput., Netw., Storage Anal.,*. IEEE, 2020, pp. 1–15.

[15] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for DC optimal power flow," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Beijing, China, 2019.

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

[17] T. Zhao, X. Pan, M. Chen, A. Venzke, and S. H. Low, "DeepOPF+: A deep neural network approach for DC optimal power flow for ensuring feasibility," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Tempe, AZ, USA, 2020, pp. 1–6.

[18] A. Velloso and P. Van Hentenryck, "Combining deep learning and optimization for preventive security-constrained DC optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3618–3628, 2021.

[19] T. Zhao, X. Pan, M. Chen, and S. Low, "Ensuring DNN Solution Feasibility for Optimization Problems with Linear Constraints," in *Int. Conf. Learn. Represent.*, 2022.

[20] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, "Data-driven optimal power flow: A physics-informed machine learning approach," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 346–354, 2020.

[21] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," in *AAAI Conf. Artif. Intell.*, vol. 34, no. 01, 2020, pp. 630–637.

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

[23] J. H. Woo, L. Wu, J.-B. Park, and J. H. Roh, "Real-time optimal power flow using twin delayed deep deterministic policy gradient algorithm," *IEEE Access*, vol. 8, pp. 213 611–213 618, 2020.

[24] S. Gupta, V. Kekatos, and M. Jin, "Controlling smart inverters using proxies: A chance-constrained DNN-based approach," *IEEE Trans. Smart Grid*, vol. 13, no. 2, pp. 1310–1321, 2022.

[25] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-hard," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 798–801, 2015.

[26] H.-D. Chiang and T. Wang, "On the existence of and lower bounds for the number of optimal power flow solutions," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1116–1126, 2018.

[27] J. Kotary, F. Fioretto, and P. Van Hentenryck, "Learning hard optimization problems: A data generation perspective," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24 981–24 992, 2021.

[28] E. Liang and M. Chen, "Generative learning for solving non-convex problem with multi-valued input-solution mapping," in *Proc. Int. Conf. Learn. Representations*, Vienna, Austria, 2024.

[29] M. Chatzos, F. Fioretto, T. W. Mak, and P. Van Hentenryck, "High-fidelity machine learning approximations of large-scale optimal power flow," *arXiv preprint arXiv:2006.16356*, 2020.

[30] E. Liang, M. Chen, and S. Low, "Low complexity homeomorphic projection to ensure neural-network solution feasibility for optimization over (non-)convex set," 2023.

[31] J. Wang and P. Srikantha, "Fast optimal power flow with guarantees via an unsupervised generative model," *IEEE Trans. Power Syst.*, pp. 1–12, 2022.

[32] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," *Proc. Int. Conf. Learn. Representations*, 2021.

[33] W. Huang and M. Chen, "DeepOPF-NGT: A fast unsupervised learning approach for solving AC-OPF problems without ground truth," in *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*, 2021.

[34] D. Owerko, F. Gama, and A. Ribeiro, "Unsupervised optimal power flow using graph neural networks," *arXiv preprint arXiv:2210.09277*, 2022.

[35] K. Chen, S. Bose, and Y. Zhang, "Unsupervised deep learning for AC optimal power flow via lagrangian duality," in *IEEE GLOBECOM*, Rio de Janeiro, Brazil, 2022, pp. 5305–5310.

[36] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 60, no. 1, pp. 150–163, 2013.

[37] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3055–A3081, 2021.

[38] M. Zhou, M. Chen, and S. H. Low, "DeepOPF-FT: One deep neural network for multiple AC-OPF problems with flexible topology," *IEEE Trans. Power Syst.*, vol. 38, no. 1, pp. 964–967, 2022.

[39] S. Liu, C. Wu, and H. Zhu, "Topology-aware Graph Neural Networks for Learning Feasible and Adaptive AC-OPF Solutions," *IEEE Trans. Power Syst.*, pp. 1–11, 2022.

[40] M. Gao, J. Yu, Z. Yang, and J. Zhao, "A physics-guided graph convolution neural network for optimal power flow," *IEEE Trans. Power Syst.*, 2023.

[41] W. A. Bukhsh, A. Grothey, K. I. M. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, 2013.

[42] R. Salgado, C. Moyano, and A. Medeiros, "Reviewing strategies for active power transmission loss allocation in power pools," *Int. J. Electr. Power Energy Syst.*, vol. 26, no. 2, pp. 81–90, 2004.

[43] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

[44] Z. Hao, H.-D. Chiang, and B. Wang, "TRUST-TECH-based systematic search for multiple local optima in deep neural nets," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–11, 2021.

[45] 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 Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

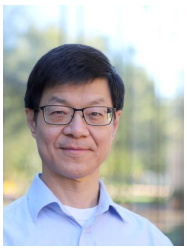
[46] K. Baker, "Emulating AC OPF solvers for obtaining sub-second feasible, near-optimal solutions," *arXiv preprint arXiv:2012.10031*, 2020.

[47] Y. Tang, K. Dvijotham, and S. Low, "Real-time optimal power flow," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2963–2973, 2017.

[48] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4909–4917, 2013.

Wanjun Huang (S'19 M'22) received the B.S. degree in electrical engineering and automation from Zhejiang University in 2016, and the Ph.D. degree from The University of Hong Kong in 2020. She was a Postdoctoral Fellow with the School of Data Science, City University of Hong Kong, Hong Kong SAR, China, and the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong SAR, China in 2021 and 2022, respectively. She is now an associate professor with the School of Automation Science and Electrical Engineering, Beihang University, China. Her research interests include voltage stability, optimal power flow, distribution network reconfiguration, and the application of machine learning in power systems.

Minghua Chen (S'04 M'06 SM' 13 F'22) received his B.Eng. and M.S. degrees from the Department of Electronic Engineering at Tsinghua University. He received his Ph.D. degree from the Department of Electrical Engineering and Computer Sciences at University of California Berkeley. He is currently a Professor of School of Data Science, City University of Hong Kong. Minghua received the Eli Jury award from UC Berkeley in 2007 (presented to a graduate student or recent alumnus for outstanding achievement in the area of Systems, Communications, Control, or Signal Processing) and The Chinese University of Hong Kong Young Researcher Award in 2013. He also received IEEE ICME Best Paper Award in 2009, IEEE Transactions on Multimedia Prize Paper Award in 2009, ACM Multimedia Best Paper Award in 2012, IEEE INFOCOM Best Poster Award in 2021, ACM e-Energy Best Paper Award in 2023, and Gradient AI Research Award in 2024. He receives the ACM Recognition of Service Award in 2017 and 2020 for the service contribution to the research community. He is currently a Senior Editor for IEEE Systems Journal, an Area Editor of ACM SIGEnergy Energy Informatics Review, and an Award Chair of ACM SIGEnergy. Minghua's recent research interests include online optimization and algorithms, machine learning in power systems, intelligent transportation systems, distributed optimization, and delay-critical networked systems. He is an ACM Distinguished Scientist and an IEEE Fellow.



Steven H. Low (Fellow, IEEE) is the F. J. Gilloon Professor of the Department of Computing & Mathematical Sciences and the Department of Electrical Engineering at Caltech. Before that, he was with AT&T Bell Laboratories, Murray Hill, NJ, and the University of Melbourne, Australia. He has held honorary/chaired professorship in Australia, China and Taiwan. He was a co-recipient of IEEE best paper awards, an awardee of the IEEE INFOCOM Achievement Award and the ACM SIGMETRICS Test of Time Award, and is a Fellow of IEEE, ACM, and CSEE. He was well-known for work on Internet congestion control and semidefinite relaxation of optimal power flow problems in smart grid. His research on networks has been accelerating more than 1TB of Internet traffic every second since 2014. His research on smart grid is providing large-scale cost effective electric vehicle charging to workplaces. He received his B.S. from Cornell and Ph.D from Berkeley, both in EE.