DeepOPF-NGT: Fast No Ground Truth Deep Learning-Based Approach for AC-OPF Problems

Wanjun Huang [1] **Minghua Chen** [1]

Abstract

AC optimal power flow (AC-OPF) problems need to be solved more frequently in the future to maintain the reliable and cost-effective operation of power systems. Recently, supervised-learning approaches have been developed to speed up the solving time of AC-OPF problems without incurring infeasibility or much optimality loss by learning the load-solution mapping embedded in the training dataset. However, it is non-trivial and computationally expensive to prepare the training dataset with single embedded mapping, due to that AC-OPF problems are non-convex and may admit multiple optimal solutions. In this paper, we develop an *unsupervised* learning approach (DeepOPF-NGT) for solving AC-OPF problems, which does not require training datasets with ground truth to operate. Instead, it uses a properly designed loss function to guide the tuning of the neural network parameters to directly learn one load-solution mapping. Preliminary results on the IEEE 30-bus test system show that the unsupervised DeepOPF-NGT approach can achieve comparable optimality, feasibility, and speedup performance against an existing supervised learning approach.

1. Introduction

With the development of the economy, electricity consumption has been increasing rapidly worldwide, and electricity generation has been one of the primary sources of carbon dioxide (CO2) emissions. According to the report by EIA (2020), CO2 emissions by the U.S. electric power sector were about 32% of total U.S. energy-related CO2 emissions. The AC optimal power flow (AC-OPF) problem has been widely studied to optimize power generation with minimum cost and all physical constraints satisfied, which is a fundamental yet challenging problem in power system operation. A study by FERC (2013) shows that an efficient AC-OPF solution approach can potentially save tens of billions of dollars every year. With more penetration of renewables, the AC-OPF problem needs to be solved more frequently to maintain the stable and economic operation of power systems. Hence, it is of great interest to solve AC-OPF problems with high efficiency.

Deep neural network (DNN)-based approach has been proposed to solve AC-OPF problems efficiently by exploiting the powerful learning ability of DNN. Much of the modern work focuses on supervised learning-based algorithms, which learns the mapping between loads and optimal generation set points based on a given dataset generated by physics-based solvers (e.g., Matpower Interior Point Solver (MIPS)). Existing approaches can be classified into two main categories: hybrid approach and stand-alone approach. The hybrid approach accelerates conventional solvers by providing warm-start points (Dong et al., 2020) or predicting active (Chen & Zhang, 2020) /inactive constraints (Hasan et al., 2021). The stand-alone approach predicts the solution of the AC-OPF problem directly without solving the optimization problem, which has a greater speedup than the hybrid approach (Pan et al., 2019; 2020a;b; Zamzam & Baker, 2020; Baker, 2020). The main idea is to predict decision variables (i.e., optimal generation set points) and then reconstruct the remaining variables using a power flow solver. Without solving power flow equations, Chatzos et al. (2020) combine DNNs and Lagrangian duality to predict all variables directly, but the critical power flow balance constraints may not be satisfied. To improve both the speedup and feasibility of existing approaches, a DNN-based voltage-constrained approach DeepOPF-V (Huang et al., 2021) is proposed, which predicts all bus voltages and then reconstructs all remaining variables via simple matrix operation.

All the above approaches require a large dataset for DNN training, which is computationally expensive, especially for large-scale power systems. More importantly, due to the non-convexity of AC-OPF problems, solvers may provide one of the locally optimal solutions for each load configura-

[1]School of Data Science, City University of Hong Kong, Hong Kong Special Administrative Region, China. Correspondence to: Wanjun Huang <wjhuang1211@gmail.com>, Minghua Chen <minghua.chen@cityu.edu.hk>.

ICML 2021 Workshop Tackling Climate Change with Machine Learning.

tion, and different initial points may even lead to different solutions (Chiang & Wang, 2018). Consequently, there is no guarantee for the optimality of the generated dataset. Perhaps more critically, the samples in the dataset may belong to different mappings between the load to (sub-)optimal solutions, which hinders the learning performance of DNNs.

To address the above issues, we take a different approach by developing a fast unsupervised learning approach named DeepOPF-NGT to solve AC-OPF problems efficiently without training dataset [1]. In this work, we use the proposed DeepOPF-NGT to find the mapping between loads and bus voltages and then directly reconstruct the remaining variables via simple scalar calculation, which guarantees the power flow balance constraints and is expected to achieve decent speedup. A loss function is properly designed to provide guidance for DNNs to find the inherent mapping between loads and AC-OPF solutions without ground truth. During training, the parameters of DNNs are updated to minimize the loss function. The preliminary results of the IEEE 30-bus test system verify the potential effectiveness of DeepOPF-NGT. Experiments for large-scale systems are currently being carried out. Note that DeepOPF-NGT provides an unsupervised learning framework for regression problems without ground truth. It can also be extended to other unsupervised regression problems.

2. Model and Methodology

2.1. AC Optimal Power Flow

Standard AP-OPF model can be formulated as follows:

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

$$\text{s.t. } P_i = \sum_{j\in\mathcal{N}} V_i V_j (g_{ij}\cos\theta_{ij} + b_{ij}\sin\theta_{ij}),\ \ i\in\mathcal{N} \tag{2}$$

$$Q_i = \sum_{j\in\mathcal{N}} V_i V_j (g_{ij}\sin\theta_{ij} - b_{ij}\cos\theta_{ij}),\ \ i\in\mathcal{N} \tag{3}$$

$$P_i = P_{gi} - P_{di},\ \ i\in\mathcal{N} \tag{4}$$

$$Q_i = Q_{gi} - Q_{di},\ \ i\in\mathcal{N} \tag{5}$$

$$\underline{P}_{gi} \le P_{gi} \le \overline{P}_{gi},\ \ i\in\mathcal{N}_G \tag{6}$$

$$\underline{Q}_{gi} \le Q_{gi} \le \overline{Q}_{gi},\ \ i\in\mathcal{N}_G \tag{7}$$

$$\underline{V}_i \le V_i \le \overline{V}_i,\ \ i\in\mathcal{N} \tag{8}$$

$$P_{ij}^2 + Q_{ij}^2 \le \overline{S}_{ij}^2,\ \ (i,j)\in\mathcal{E} \tag{9}$$

$$P_{ij} = g_{ij}V_i^2 - V_iV_j(b_{ij}\sin\theta_{ij} + g_{ij}\cos\theta_{ij}), (i,j)\in\mathcal{E} \tag{10}$$

$$Q_{ij} = -b_{ij}V_i^2 - V_iV_j(g_{ij}\sin\theta_{ij} - b_{ij}\cos\theta_{ij}), (i,j)\in\mathcal{E} \tag{11}$$

$$\underline{\theta}_{ij} \le \theta_{ij} \le \overline{\theta}_{ij},\ \ (i,j)\in\mathcal{E} \tag{12}$$

where $\mathcal{N}$, $\mathcal{N}_G$ and $\mathcal{E}$ denote the sets of all buses, generation buses and transmission lines; g_{ij} and b_{ij} represent conductance and susceptance of branch (i,j), respectively. For bus i, P_i and Q_i represent net active and reactive power injections, respectively; P_{gi}, Q_{gi}, P_{di} and Q_{di} denote active power generation, reactive power generation, active load and reactive load, respectively; V_i and θ_i are voltage magnitude and angle, respectively. For branch (i,j), $\theta_{ij} = \theta_i - \theta_j$ denotes branch angle; P_{ij} and Q_{ij} are active and reactive branch power flows, respectively. The upper and lower bounds of variable x are denoted by $\overline{x}$ and $\underline{x}$, respectively. The AC-OPF problem aims to minimize generation costs in (1) with the satisfaction of all physical constraints in (2)-(12). The Kirchhoff's circuit laws are ensured by (2)-(3); net power injections are derived by (4)-(5); active and reactive power generation limits are enforced by (6)-(7); voltage magnitude limit is ensured by (8); branch flow is restricted by (17)-(11); and voltage angles are restricted by (12).

2.2. Unsupervised Learning Approach without Ground Truth

The schematic of the proposed DeepOPF-NGT is illustrated in Figure 1. This DNN-based model is comprised of fully connected layers with rectified linear unit (ReLU) activation function on each hidden layer and sigmoid activation function on the output layer. It aims to learn the mapping between load configurations $(\boldsymbol{P_d}, \boldsymbol{Q_d})$ and bus voltages $(\boldsymbol{\theta}, \boldsymbol{V})$, where $\boldsymbol{P_d}$ and $\boldsymbol{Q_d}$ are vectors of active and reactive loads, respectively; $\boldsymbol{\theta}$ and $\boldsymbol{V}$ are vectors of voltage angles and magnitudes, respectively. Using the well-trained DNNs, the predicted voltage magnitudes $\hat{\boldsymbol{V}}$ and voltage angles $\hat{\boldsymbol{\theta}}$ can be obtained instantly with the input of $(\boldsymbol{P_d}, \boldsymbol{Q_d})$. Then, using the predicted voltage magnitudes $\hat{\boldsymbol{V}}$, voltage angles $\hat{\boldsymbol{\theta}}$ and the load input $(\boldsymbol{P_d}, \boldsymbol{Q_d})$, we can easily compute the right-hand side (RHS) of the equations in (2)-(3). Then, the remaining solution variables $\hat{\boldsymbol{P}}_{\boldsymbol{g}}$, $\hat{\boldsymbol{Q}}_{\boldsymbol{g}}$ and some auxiliary variables $(\hat{\boldsymbol{P}}_{\boldsymbol{d}}, \hat{\boldsymbol{Q}}_{\boldsymbol{d}})$ are directly calculated from (4)-(5) using the obtained RHS values without the need to solve non-linear power flow equations. Specifically, for each

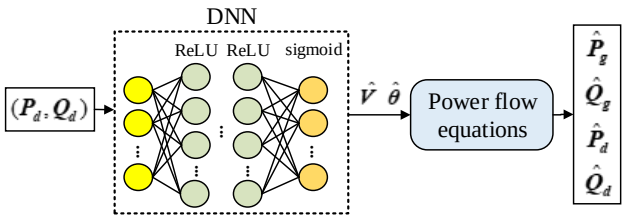

Figure 1. Schematic of the proposed no ground truth deep-learning based approach DeepOPF-NGT.

[1] Recently, an unsupervised learning framework for AC-OPF problems was also developed by Donti et al.(2021). However, it may not guarantee the critical power flow balance constraints.

bus i, 1) if there are only generators or loads, its predicted active/reactive generation (i.e., $\hat{P}_{gi}$/$\hat{Q}_{gi}$) or active/reactive load (i.e., $\hat{P}_{di}$/$\hat{Q}_{di}$) is 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 (2)-(5). The objective function is calculated by (1) after obtaining $\hat{\boldsymbol{P}}_{\boldsymbol{g}}$. Due to the voltage prediction errors, there could be unsatisfied loads, i.e., the mismatches between ($\boldsymbol{P_d}, \boldsymbol{Q_d}$) and ($\hat{\boldsymbol{P}}_{\boldsymbol{d}}$, $\hat{\boldsymbol{Q}}_{\boldsymbol{d}}$), which will be discussed in Section. 2.3.

To guide the unsupervised training of DNNs without ground truth, the loss function is designed as follows:

$$\mathcal{L} = k_{gen}\mathcal{L}_{gen} + \mathcal{L}_{cons} + k_d\mathcal{L}_d, \tag{13}$$

where k_{gen} and k_d positive constants. In the loss function $\mathcal{L}$, $\mathcal{L}_{gen}$ is designed to minimize the generation cost, $\mathcal{L}_{cons}$ to find feasible solutions satisfying constraints in (2)-(12), and $\mathcal{L}_d$ to satisfy demanded loads. The power flow balance constraints in (2)-(3) are satisfied automatically since power injections can always be obtained with predicted bus voltages. Hence, by applying $\mathcal{L}$, there is no need of ground truth to train DNNs. The generation cost $\mathcal{L}_{gen}$ is the generation cost calculated by

$$\mathcal{L}_{gen} = \sum_{i\in\mathcal{N}_G} (a_i P_{g,i}^2 + b_i|P_{g,i}| + c_i), \tag{14}$$

where a_i, b_i and c_i are positive constants. The term $\mathcal{L}_{cons}$ is the penalty for constraint violation during training, which is obtained by

$$\mathcal{L}_{cons} = k_g\mathcal{L}_g + k_s\mathcal{L}_{S_l} + k_\theta\mathcal{L}_{\theta_l}, \tag{15}$$

where k_g, k_s and k_θ are positive constants; $\mathcal{L}_g$, $\mathcal{L}_{S_l}$ and $\mathcal{L}_{\theta_l}$ are penalties for the violation of generation, branch flow and branch angle constraints during training, respectively, which are computed as below:

$$\begin{aligned}\mathcal{L}_g = \sum_{i\in\mathcal{N}_G} &[\max(\hat{P}_{gi} - \overline{P}_{gi}, 0) + \max(\underline{P}_{gi} - \hat{P}_{gi}, 0) \\ &+ \max(\hat{Q}_{gi} - \overline{Q}_{gi}, 0) + \max(\underline{Q}_{gi} - \hat{Q}_{gi}, 0)]\end{aligned} \tag{16}$$

$$\mathcal{L}_{s_l} = \sum_{i,j\in\mathcal{N}} [\max(\hat{S}_{ij} - \overline{S}_{ij}, 0)] \tag{17}$$

$$\mathcal{L}_{\theta_l} = \sum_{i,j\in\mathcal{N}} [\max(\hat{\theta}_{ij} - \overline{\theta}_{ij}, 0) + \max(\underline{\theta}_{ij} - \hat{\theta}_{ij}, 0)] \tag{18}$$

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

$$\mathcal{L}_d = \sum_{i\in\mathcal{N}_L} (|\hat{P}_{di} - P_{di}| + |\hat{Q}_{di} - Q_{di}|). \tag{19}$$

As discussed in (Huang et al., 2021), due to prediction errors, there could be mismatches between the given loads and those obtained from predicted voltages. However, the unsatisfied loads are also inevitable in conventional approaches. In addition, considering power losses in transmission lines, a small ratio (around 1%) of load-generation imbalance is acceptable.

2.3. Neural Network Training for DeepOPF-NGT

The DNNs of DeepOPF-NGT is trained by minimizing the loss function $\mathcal{L}$. A simple and viable approach is to apply the gradient descent algorithm. Denote the parameters of DNNs as ϕ and the corresponding mapping from the input $\boldsymbol{x} = (\boldsymbol{P_d}, \boldsymbol{Q_d})^T$ to the output $\boldsymbol{y} = (\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{V}})$ as $\boldsymbol{y} = f(\boldsymbol{x}, \phi)$. Then, we can obtain $\hat{\boldsymbol{P}}_{\boldsymbol{g}}(f(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{Q}}_{\boldsymbol{d}}(f(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{P}}_{\boldsymbol{d}}(f(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{Q}}_{\boldsymbol{d}}(f(\boldsymbol{,}\phi))$, $\hat{\boldsymbol{Q}}_{\boldsymbol{d}}(f(\boldsymbol{x}, \phi))$, $\hat{\boldsymbol{S}}_{\boldsymbol{ij}}(f(\boldsymbol{x}, \phi))$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{ij}}(f(\boldsymbol{x}, \phi))$. Thus, ϕ is updated according to

$$\phi_{t+1} = \phi_t - \alpha_t \nabla_{\phi_t}\mathcal{L}, \tag{20}$$

where α_t is a positive step size at the t-th epoch of training, and the gradient $\nabla_\phi\mathcal{L}$ can be obtained by using Chain rule as below

$$\begin{aligned}\nabla_\phi\mathcal{L} &= \nabla_{\boldsymbol{y}}\mathcal{L} \cdot \nabla_\phi \boldsymbol{y} \\ &= (k_{gen}\nabla_{\boldsymbol{y}}\mathcal{L}_{gen} + \nabla_{\boldsymbol{y}}\mathcal{L}_{cons} + k_d\nabla_{\boldsymbol{y}}\mathcal{L}_d) \cdot \nabla_\phi \boldsymbol{y} \\ &= [(k_{gen}\nabla_{\hat{\boldsymbol{P}}_{\boldsymbol{g}}}\mathcal{L}_{gen} + k_g\nabla_{\hat{\boldsymbol{P}}_{\boldsymbol{g}}}\mathcal{L}_g + k_{s_l}\nabla_{\hat{\boldsymbol{P}}_{\boldsymbol{g}}}\mathcal{L}_{s_l}) \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{P}}_{\boldsymbol{g}}(\boldsymbol{y}) \\ &+ (k_g\nabla_{\hat{\boldsymbol{Q}}_{\boldsymbol{g}}}\mathcal{L}_g + k_{s_l}\nabla_{\hat{\boldsymbol{Q}}_{\boldsymbol{g}}}\mathcal{L}_{s_l}) \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{Q}}_{\boldsymbol{g}}(\boldsymbol{y}) \\ &+ k_d\nabla_{\hat{\boldsymbol{S}}_{\boldsymbol{ij}}}\mathcal{L}_{s_l} \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{S}}_{\boldsymbol{ij}}(\boldsymbol{y}) + k_\theta\nabla_{\boldsymbol{\theta}_{\boldsymbol{ij}}}\mathcal{L}_{\theta_l} \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{\theta}}_{\boldsymbol{ij}}(\boldsymbol{y}) \\ &+ k_d\nabla_{\hat{\boldsymbol{P}}_{\boldsymbol{d}}}\mathcal{L}_d \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{P}}_{\boldsymbol{d}}(\boldsymbol{y}) + k_d\nabla_{\hat{\boldsymbol{Q}}_{\boldsymbol{d}}}\mathcal{L}_d \cdot \nabla_{\boldsymbol{y}}\hat{\boldsymbol{Q}}_{\boldsymbol{d}}(\boldsymbol{y})] \cdot \nabla_\phi \boldsymbol{y}.\end{aligned} \tag{21}$$

For faster and better convergence, the mini-batch stochastic gradient descent is employed to calculate $\nabla_{\boldsymbol{y}}\mathcal{L}$, which makes a trade-off between batch gradient descent and stochastic gradient descent (Hinton et al.).

2.4. Discussion

In this work, the optimal parameters of DNNs are searched by using mini-batch gradient descent algorithm. On one hand, due to the non-convexity of loss function $\mathcal{L}$, the obtained parameters of DNNs may be sub-optimal. Hence, the predicted AC-OPF solutions may be sub-optimal. However, existing solvers can only provide sub-optimal solutions as well. Up till now, how to find globally optimal solutions for AC-OPF under general settings is still an open problem. On the other hand, compared with existing supervised learning-based approaches, the proposed DeepOPF-NGT does not need ground truth, and it has the potential to find better solutions than conventional solvers. Moreover, as discussed in Section 1, due to that AC-OPF problems are

Table 1. Parameter settings for DeepOPF-NGT and DeepOPF-V.

APPROACH	BATCH SIZE	LEARNING RATE	EPOCH	HIDDEN LAYERS
DEEPOPF-NGT	100	0.001	300	128-64-64
DEEPOPF-V	100	0.001	1000	128-128-64

Table 2. Performance comparison results in IEEE 30-bus system.

METRIC	DEEPOPF-NGT	DEEPOPF-V
$\eta_{opt}(\%)$	<0.4	<0.1
$\eta_{\boldsymbol{V}}(\%)$	100.0	100.0
$\eta_{\boldsymbol{P_g}}(\%)$	100.0	100.0
$\eta_{\boldsymbol{Q_g}}(\%)$	100.0	100.0
$\eta_{\boldsymbol{S_l}}(\%)$	100.0	100.0
$\eta_{\boldsymbol{\theta_l}}(\%)$	100.0	100.0
$\eta_{\boldsymbol{P_d}}(\%)$	99.3	99.8
$\eta_{\boldsymbol{Q_d}}(\%)$	99.2	99.3
η_{sp}	AROUND$\times$640	AROUND$\times$610

non-convex and may have multiple optimal solutions, the load-solution pairs in the training dataset may correspond to multiple load-solution mappings. This creates difficulty for supervised learning approaches that were designed to learn just one mapping from the dataset. In contrast, our unsupervised learning approach can avoid this issue.

3. Case Study

3.1. Experimental Setup

Simulations are conducted on IEEE 30-bus test system. The dataset contains 12,500 load samples with an 80–20% training-test split. Each sample only contains a set of load scenarios sampled randomly for each load bus from a uniform distribution of 10% variation around the default load. To verify its effectiveness, the proposed DeepOPF-NGT is compared with DeepOPF-V (Huang et al., 2021) which is a supervised learning approach that learns the mapping between loads and bus voltages. The dataset for DeepOPF-V contains not only 12,500 different sets of load scenarios mentioned above but also the corresponding ground truths, i.e., AC-OPF solutions obtained by the OPF solver MIPS.

The DNN-based models are designed on the platform of Pytorch. The hyper-parameters are fine-tuned by trial and error (see Table. 1). Note that the size of the hidden layer in DeepOPF-NGT is larger than that in DeepOPF-V. The reason is that DeepOPF-NGT predicts voltage magnitudes and angles together in one single DNN-based model. In contrast, while DeepOPF-V predicts voltage magnitudes and angles separately in two DNN-based models. Hence, the output dimension of the DNNs in DeepOPF-NGT is larger than that in DeepOPF-V. Simulations are run on the quad-core (i7-3770@3.40G Hz) CPU workstation with 16GB RAM.

3.2. Performance Evaluation

To evaluate the performance of the proposed DeepOPF-NGT comprehensively, the following metrics are considered:

1) ***Speedup Factor***: The speedup factor η_{sp} measures the average ratios of the computation time consumed by MIPS to solve the original AC-OPF to the computation time consumed by DeepOPF-NGT. A larger value of η_{sp} refers to better speedup performance.

2) ***Optimality Loss***: It measures the average relative deviation η_{opt} between the optimal objective value found by MIPS and that by DeepOPF-NGT.

3) ***Constraint Satisfaction***: It evaluates the feasibility of predicted solutions by the percentage of bound constraints satisfied. 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.

4) ***Load Satisfaction Ratio***: It is defined as the percentage of demanded loads satisfied. The active and reactive load satisfaction ratios are denoted as $\eta_{\boldsymbol{P}_d}$ and $\eta_{\boldsymbol{Q}_d}$, respectively.

The experimental results are summarized in Table. 2. As seen, the optimality losses of these two approaches are both less than 0.4% with inequality constraints all satisfied. Besides, the load satisfaction ratios are all larger than 99%. Both of these two approaches can speed up the solution of AC-OPF by more than three orders of magnitude, i.e., around$\times$640 and around$\times$610 for DeepOPF-GNT and DeepOPF-V, respectively. Note that the proposed DeepOPF-NGT has a little larger speedup than DeepOPF-V. As illustrated in Section 3.1, DeepOPF-NGT has a smaller size of DNNs than that of DeepOPF-V, resulting in a little less computation time than DeepOPF-V. Note that the parallel computation time was recorded for DeepOPF-V.

Overall, in the small-scale simulations above, the unsupervised learning approach DeepOPF-NGT has comparable performance with the supervised learning-based approach but does not need to prepare a large dataset for DNN training. We are currently carrying out extensive simulations for larger systems and higher load variation.

4. Conclusion

We propose a fast *unsupervised* learning approach DeepOPF-NGT to solve AC-OPF efficiently without ground truth. It predicts bus voltages using the well-trained DNN-

based model and then reconstructs all remaining variables according to power flow equations. Compared with existing approaches that only learn the load-solution mapping embedded in the training dataset, the proposed DeepOPF-NGT directly identifies the inherent load-solution mapping without ground truth. A loss function based on the AC-OPF problem is properly designed to guide the DNN training with mini-batch stochastic gradient descent. Simulation results on the IEEE 30-bus test system show that DeepOPF-NGT has comparable performance with the state-of-the-art supervised learning-based approach. It provides feasible solutions with negligible optimality loss less than 0.4% and decent computation speedup that is more than three orders of magnitude faster than conventional solver MIPS.

In this paper, we only present very preliminary results to show the potential of DeepOPF-NGT. We are also actively carrying out experiments for large-scale systems. Besides, we will explore a more efficient algorithm for DNN training. There is a great potential that the proposed DeepOPF-NGT may find better solutions than conventional solvers since its performance will not be restricted by the dataset generated by OPF solvers. Moreover, the scalability of DeepOPF-NGT for large-scale power systems will also be explored.

References

Baker, K. Emulating ac opf solvers for obtaining sub-second feasible, near-optimal solutions. *arXiv preprint arXiv:2012.10031*, 2020.

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

Chen, Y. and Zhang, B. Learning to solve network flow problems via neural decoding. *arXiv preprint arXiv:2002.04091*, 2020.

Chiang, H.-D. and Wang, T. On the existence of and lower bounds for the number of optimal power flow solutions. *IEEE Transactions on Power Systems*, 34(2):1116–1126, 2018.

Dong, W., Xie, Z., Kestor, G., and Li, D. Smart-pgsim: using neural network to accelerate ac-opf power grid simulation. *arXiv preprint arXiv:2008.11827*, 2020.

Donti, P. L., Rolnick, D., and Kolter, J. Z. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*, 2021.

EIA. Monthly energy review. Technical report, U.S. Department of Energy, Washington, DC, 2020.

Hasan, F., Kargarian, A., and Mohammadi, J. Hybrid learning aided inactive constraints filtering algorithm to enhance ac opf solution time. *Trans. Ind. Applicat.*, 2021.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.

Huang, W., Pan, X., Chen, M., and Low, S. H. Deepopf-v: Solving ac-opf problems efficiently. *arXiv preprint arXiv:2103.11793*, 2021.

Pan, X., Zhao, T., and Chen, M. Deepopf: Deep neural network for dc optimal power flow. In *IEEE SmartGridComm*, 2019.

Pan, X., Chen, M., Zhao, T., and Low, S. H. Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems. *arXiv preprint arXiv:2007.01002*, 2020a.

Pan, X., Zhao, T., Chen, M., and Zhang, S. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. *IEEE Trans. Power Syst.*, 2020b.

Schecter, A. and O'Neill, R. P. Exploration of the acopf feasible region for the standard ieee test set. *FERC Staff Technical Paper*, 2013.

Zamzam, A. S. and Baker, K. Learning optimal solutions for extremely fast ac optimal power flow. In *IEEE SmartGridComm*, 2020.