DeepOPF-FT: One Deep Neural Network for Multiple AC-OPF Problems with Flexible Topology

Min Zhou, Minghua Chen, *Fellow, IEEE*, and Steven H. Low, *Fellow, IEEE*

***Abstract*—We propose DeepOPF-FT as an embedded training approach to design *one* deep neural network (DNN) for solving multiple AC-OPF problems with flexible topology and line admittances, addressing a critical limitation of learning-based OPF schemes. The idea is to embed the discrete topology representation into the continuous admittance space and train a DNN to learn the mapping from (load, admittance) to the corresponding OPF solution. We then employ the trained DNN to solve AC-OPF problems over any power network with the same bus, generation, and line capacity configurations but different topology and/or line admittances. Simulation results over IEEE 9-/57- bus and a synthetic 2000-bus test cases demonstrate the effectiveness of our design and highlight the training efficiency improvement of DeepOPF-FT over training one DNN for every combination of power network topology and line admittances.**

***Index Terms*—Optimal power flow; deep neural network**

I. Introduction

Recently, there has been growing interest in employing machine learning, in particular deep neural network (DNN), to directly solve the optimal power flow (OPF) problems, in a fraction of the time solved by iterative solvers. The idea is to leverage the approximation capability of DNN to learn the load-solution mapping of the OPF problem [1]. Then one can feed load to the DNN to instantly obtain a solution. To date, a number of studies have shown that DNNs can generate quality solutions for various OPF formulations with a few orders of magnitude speedup as compared to iterative solvers [1]–[12].

A key limitation of existing DNN methods is that the trained DNN is only applicable for solving OPF problems over a specific system topology and line admittance. When the topology or admittances change, one needs to retrain the DNN to learn a new load-solution mapping. Retraining DNNs in real-time [9], [13], or pre-training multiple DNNs offline for all possible combinations of topology and admittances, incurs significant computational and data complexity and may not be practical.

In this paper, we propose DeepOPF-FT as an embedded training approach to train one DNN for solving multiple AC-OPF problems with flexible topology and admittances, without retraining. We embed discrete topology representation in continuous admittances and train a DNN to learn the mapping from (load, admittance) to the AC-OPF solution. Simulation results over modified IEEE 9-/57- bus and synthetic 2000-bus test cases show that DeepOPF-FT generates AC-OPF solutions with up to 0.92% optimality loss and at least 95% feasibility rate, over power networks with the same bus, generator, and line capacity configurations but different topology and/or admittance. Our results also highlight the training efficiency of DeepOPF-FT over employing one DNN for every combination of topology and line admittance. To the best of our knowledge, DeepOPF-FT is the first work that trains one DNN for solving multiple AC-OPF problems with flexible topology and admittance, without retraining.

The work presented in this paper was supported in part by a General Research Fund from Research Grants Council, Hong Kong (Project No. 11203122), an InnoHK initiative, The Government of the HKSAR, and Laboratory for AI-Powered Financial Technologies, NSF through award ECCS 1931662, and C3.ai/UC Berkeley through award No. 11015.

M. Zhou and M. Chen are with School of Data Science, City University of Hong Kong. S. Low is with Computing and Mathematical Sciences and Electrical Engineering, Caltech. Corresponding author: Minghua Chen.

II. The AC-OPF problem and topology embedding

The standard AC-OPF problem is formulated as

$$\min \sum\nolimits_{i\in\mathcal{B}} C(p_i^g) \tag{1}$$

$$\text{s.t. } \underline{p_i^g} \le p_i^g \le \overline{p_i^g},\ \ \underline{q_i^g} \le q_i^g \le \overline{q_i^g},\ i\in\mathcal{B}, \tag{2}$$

$$\underline{v_i} \le v_i \le \overline{v_i},\ i\in\mathcal{B}, \tag{3}$$

$$\underline{\theta_{ij}} \le \theta_{ij} = \theta_i - \theta_j \le \overline{\theta_{ij}},\ (i,j)\in\mathcal{L}, \tag{4}$$

$$p_{ij}^f = g_{ij}v_i^2 - v_iv_j(g_{ij}\cos\theta_{ij} + b_{ij}\sin\theta_{ij}), (i,j)\in\mathcal{L}, \tag{5}$$

$$q_{ij}^f = -b_{ij}v_i^2 - v_iv_j(g_{ij}\sin\theta_{ij} - b_{ij}\cos\theta_{ij}), (i,j)\in\mathcal{L}, \tag{6}$$

$$p_i^g - p_i^d = \sum\nolimits_{(i,j)\in\mathcal{L}} p_{ij}^f,\ i\in\mathcal{B}, \tag{7}$$

$$q_i^g - q_i^d = \sum\nolimits_{(i,j)\in\mathcal{L}} q_{ij}^f,\ i\in\mathcal{B}, \tag{8}$$

$$(p_{ij}^f)^2 + (q_{ij}^f)^2 \le (\overline{s_{ij}})^2,\ (i,j)\in\mathcal{L}, \tag{9}$$

$$\text{var. } p_i^g, q_i^g, v_i, \theta_i, i\in\mathcal{B}.$$

$\mathcal{B}$ and $\mathcal{L}$ represent the sets of buses and branches, respectively. g_{ij} and b_{ij} are the conductance and susceptance of line (i,j), respectively. p_i^g, q_i^g, p_i^d and q_i^d are the active and reactive power generation, active and reactive load at bus i, respectively. v_i and θ_i denote the voltage magnitude and angle at bus i, respectively. $\overline{x}$ and $\underline{x}$ denote the upper and lower bound of variable x, respectively. $\overline{s_{ij}}$ is the branch flow limit of line (i,j). Constraints (5)-(6) define active and reactive line flows and (7)-(8) ensure power flow balance at every bus. The generation limits are given by (2). Voltage magnitude and phase angle constraints are specified in (3)-(4). The branch flow limit is enforced by (9). The objective in (1) is to minimize the total quadratic active power generation cost.

For the above AC-OPF formulation, we assume some or all of transmission lines have switches and can be switched on or off based on operation conditions or on contingency. Switching a line (i,j) with given capacity on (resp. off) is equivalent to adjusting the corresponding b_{ij} and g_{ij} from zero to non-zero values (resp. from non-zero values to zero). Thus, the topology,

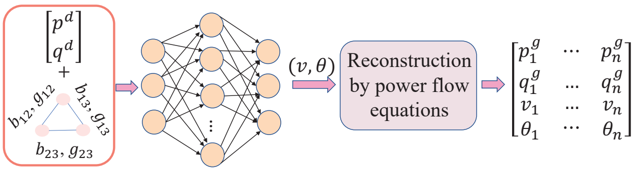

Fig. 1. Schematic of DeepOPF-FT. A DNN is used to learn the mapping from ($\boldsymbol{p}^d, \boldsymbol{q}^d$, $\boldsymbol{b}, \boldsymbol{g}$) to ($\boldsymbol{v}$, $\boldsymbol{\theta}$). The remaining variables, i.e., ($\boldsymbol{p}^g$, $\boldsymbol{q}^g$), are computed by using the power flow equations.

i.e., whether lines (with pre-specified capacities) are on or off, can be embedded in the values of line admittances.

Next, we will design *one* DNN to solve multiple AC-OPF problems over power networks with the same bus, generator, and line capacity configuration, but different topology and/or line admittances. Such a *flexible* topology setting can model the situation with time-varying line admittances or line switching-based contingency or network reconfiguration [14].

III. DeepOPF-FT: Embedded Training for Solving AC-OPF Problems with Flexible Topology

The schematic of DeepOPF-FT is shown in Fig. 1. It embeds the discrete topology representation into the continuous admittance and employs a DNN to learn the mapping from (load, admittance) to bus voltages in the AC-OPF solution. We adopt the multi-layer feed-forward DNN structure, in which the ReLU activation function is used in the hidden layers. The loss function is the total mean squared error between DNN prediction and the ground truth. DeepOPF-FT uses the trained DNN to first obtain bus voltages, i.e., $\boldsymbol{v}$ and $\boldsymbol{\theta}$, and then reconstruct the remaining variables, i.e., ($\boldsymbol{p^g}$, $\boldsymbol{q^g}$), by simple computation using the power flow equations. Such a predict-and-reconstruct mechanism [1] guarantees in-network power flow equality constraints and reduces the number of variables to be predicted by DNN. Finally, we apply the post-processing technique in [1] to further improve the feasibility of the obtained solutions.

Discussion: (i) Solving AC-OPF problems with flexible topology and admittance is challenging for learning-based methods. While it is possible for system operators to train one DNN per power network, the sampling complexity can be extraordinarily high, as we observe in Sec. IV. Another possible approach is discrete training, which learns the mapping from (discrete topology, load) to the corresponding OPF solutions. However, as discrete training learns a discrete mapping from (discrete topology, load) to OPF solutions, it requires significantly more data to train the DNN, and even so the obtained DNN may not generalize well, as we observe in Sec. IV. Further, discrete training does not consider the setting with flexible (continuous) admittance. In our simulation, we use discrete training as a baseline for comparison. (ii) Due to the prediction error of voltages, there could be mismatches between power injection and load demand in buses. We note that injection-demand mismatches are also inevitable in iterative solvers [1]. System operators can use distributed controllable energy sources, such as electric batteries, to fully satisfy the load.

TABLE I
Parameter settings and data split.

# Bus	DNN structure	Batch size	Training epoch	Learning rate	# Training/test data
57	278/1024/512/256	128	5350	1e-4	50K/50K
9	42/1024/512/256	256	2650	1e-4	100K/125K
2000	4060/2048/2048/2048	512	4500	1e-5	50K/12.5K

TABLE II
Performance comparison over the modified IEEE 57-bus system.

Metric	DeepOPF-FT (50,000)	DIS-V1 (50,000)	DIS-V2 (50,000)	DIS-V1 (150,000)	DIS-V2 (150,000)
η_{opt} (%)	0.14	-4.29	-1.31	-4.79	1.07
η_v/η_θ (%)	-	-	-	-	-
η_{p^g} (%)	95.0	94.3	93.3	97.0	96.1
η_{q^g} (%)	96.0	92.4	95.6	96.3	94.0
η_{sl} (%)	>99.9	>99.9	>99.9	>99.9	>99.9
η_{p^d} (%)	97.2	92.7	95.2	95.8	95.8
η_{q^d}(%)	94.3	87.5	91.2	93.0	91.6
η_{sp}	×129	×130	×132	×130	×130

IV. Numerical Experiments

We test DeepOPF-FT over small 9-bus, medium 57-bus, and large 2000-bus systems; see the configurations and our codes in [15]. We conduct simulations using a quad-core (i7-3770@3.40G Hz) CPU workstation with 16GB RAM. Table I gives the DNN structure, training parameters, and training/testing dataset sizes. We follow a common approach for DNN training and performance evaluation: (i) *randomly* sample from the load region to construct data points with labels/ground-truths, (ii) split the data into the training set and testing set, (iii) use the training set to train the DNN, and (iv) use the testing set to evaluate its performance.

More specifically, we sample load uniformly at random in [80%, 120%] of its default value in training/test datasets. In the training dataset, we sample admittances uniformly at random in two regions covering the on/off status of lines: (i) the region for off status where admittances are within [-2%, 2%] of default admittances and (ii) the region for on status where admittances are within [2%, 120%] of default admittances, with probability 3.3% and 96.7%, respectively. In the test dataset and baseline methods, we set two scenarios for admittances: (i) fixed admittance where admittances are set as defaults admittances and (ii) flexible admittance where admittances are sampled uniformly at random in [80%, 120%] of default admittances to capture the slight admittance variation in real power systems. We obtain the ground truths using the primal-dual interior-point method in the MATPOWER Interior Point Solver (MIPS) [16], which is able to generate close-to-optimal solutions to AC-OPF [17]. Therefore, our DNN learns the mapping from (load, admittance) to MIPS solutions. We implement the DNN schemes using Pytorch, which is based on Python. The reference solver MIPS is also based on Python.

We use the following metrics to evaluate the performance of DeepOPF-FT and baselines: (i) **Optimality loss:** The optimality loss η_{opt} evaluates the average relative difference of the objective values obtained by DeepOPF-FT and the ground truth. Closer to zero is better. (ii) **Constraint satisfaction:** It measures the average percentage of inequality constraint

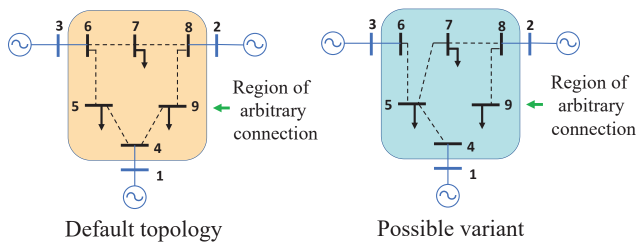

Fig. 2. Topology variation in the IEEE 9-bus system.

TABLE III
PERFORMANCE COMPARISON IN THE MODIFIED IEEE 9-BUS SYSTEM.

Metric	DeepOPF-FT		DeepOPF-V for single topology		
	(FT, –)	(FT, FA)	(FT, –)	(FT, FA)	(–, –)
η_{opt} (%)	0.84	0.92	94.60	95.23	-0.95
η_v/η_θ (%)	-	-	-	-	-
η_{p^g} (%)	> 99.9	>99.9	53.6	53.6	100
η_{q^g} (%)	> 99.9	100	97.8	97.8	> 99.9
η_{sl} (%)	>99.9	>99.9	96.6	96.3	100
η_{p^d} (%)	97.4	97.3	74.8	74.6	97.0
η_{q^d} (%)	95.3	95.0	57.0	56.8	91.8
η_{sp}	×124	×122	×88	×86	×133

satisfaction, including the constraint satisfaction ratio of active power generation (η_{p^g}), reactive power generation (η_{q^g}), voltage magnitude (η_v), phase angle difference (η_θ), and branch flow limits (η_{sl}). Closer to 100 is better. (iii) **Speedup:** The speedup factor η_{sp} is the average ratio between the running time of MIPS and DeepOPF-FT. It measures the speedup gain of DeepOPF-FT over MIPS. Higher is better. (iv) **Load satisfaction:** The load satisfaction ratio is the average percentage of the satisfied loads. The load satisfaction ratio for active load and reactive load are denoted as η_{p^d} and η_{q^d}, respectively. Closer to 100 is better.

Performance on topology reconfiguration: We carry out simulations over the modified 57-bus test system with 14 configurable lines. Discrete training with fixed admittance (DIS-V1) [10] and flexible admittance (DIS-V2) are baselines. For fair comparison, when the admittances are in the region for off status in DeepOPF-FT, we disconnect the corresponding line in discrete training. Otherwise, we sampled the corresponding admittances to be fixed in DIS-V1 and be flexible in DIS-V2. The test data are based on the N-4/5/6 contingency (each account for 1/3 of the data) with flexible admittances.

Table II shows that DeepOPF-FT achieves better performance in optimality, feasibility, and load satisfaction than DIS-V1 and DIS-V2 schemes. It also shows that the DIS-V1 and DIS-V2 schemes need 3x the amount of training data to achieve performance comparable to DeepOPF-FT, highlighting its advantage in training efficiency.

Performance over arbitrary topology: We evaluate the performance of DeepOPF-FT in solving AC-OPF problems over arbitrary topology in the modified IEEE 9-bus[1] test system. As shown in Fig. 2, all 15 lines incident on buses No. 4 – 9 can be switched on/off. We select 19,647 out of the 32,768 possible topologies that support the same load region.

The performance of DeepOPF-FT is evaluated over two scenarios: (i) flexible topology but fixed admittances ((FT, –)) and (ii) flexible topology and flexible admittances ((FT, FA)). We also select 10 topologies randomly from the 19,647 topologies, train, and evaluate DeepOPF-V [1] for each topology (represented by fixed topology and fixed admittance ((–, –))) as the baseline. Table III shows that DeepOPF-FT achieves much better performance in optimality, feasibility and load satisfaction over all possible testing topologies, as compared to DeepOPF-V. This shows (i) DNN trained for one topology does not work well over other topologies, and (ii) the effectiveness of the embedded training design. DeepOPF-V needs 300[2] training data to achieve the shown performance for a single topology. This suggests that 300×19647 training data will be needed if we train one DeepOPF-V for every possible topology with comparable performance.

Performance on a large-scale system: We carry out simulations over a 2000-bus test system to show the scalability of DeepOPF-FT. We test the performance of DeepOPF-FT in the default topology (N) and N-1 contingency (N-1). Table IV shows (i) DeepOPF-FT achieves satisfactory performance in optimality, feasibility and load satisfaction over the two test scenarios, suggesting its scalibility to large test systems, and (ii) DeepOPF-V achieves comparable performance to DeepOPF-FT, indicating the insensitivity of OPF solutions to N-1 contingency in the large 2000-bus test system. We also observe that DeepOPF-FT achieves lower speedup than DeepOPF-V, as DeepOPF-FT employs a larger DNN for learning a higher dimensional mapping in its design.

TABLE IV
PERFORMANCE COMPARISON IN THE 2000-BUS SYSTEM.

Metric	DeepOPF-FT		DeepOPF-V for single topology	
	N	N-1	N	N-1
η_{opt} (%)	0.02	0.11	0.28	0.27
η_v/η_θ (%)	-	-	-	-
η_{p^g} (%)	>99.9	>99.9	>99.9	>99.9
η_{q^g} (%)	99.9	99.8	99.9	99.9
η_{sl} (%)	>99.9	>99.9	>99.9	>99.9
η_{p^d} (%)	98.9	98.8	99.4	99.4
η_{q^d} (%)	95.9	95.5	96.1	96.1
η_{sp}	×7646	×7743	×16335	×16083

V. CONCLUDING REMARK

To our best knowledge, DeepOPF-FT is the first that trains one DNN for solving multiple AC-OPF problems under the same bus, generator, and line capacity configuration, but with different topology and line admittances. Simulation results show that it achieves better optimality, feasibility, and load satisfaction performance than training one DNN for every combination of topology and admittance.

We discuss the limitations of this study and future directions in the following. (i) As compared to training one DNN over

[1]We note that for AC-OPF problems over small-scale cases, even minor changes in topology or admittances lead to notable differences in the solution, making it challenging to train one DNN to work effectively over different (topology, admittance) combinations. Focusing on small-scale test cases also allows us to evaluate the performance of DeepOPF-FT over *all* possible topologies with the same bus, generation, and line capacity configurations.

[2]We note that setting the training size to be 300 is reasonable to DNN-based AC-OPF solvers for small-scale systems [6].

a specific power network and line admittance, DeepOPF-FT may require a larger DNN size for learning the higher dimensional (load, admittance) to solution mapping. (ii) This study focuses on solving the standard AC-OPF problem. It is an interesting direction to extend the approach to AC-OPF problems considering emerging technologies, e.g., energy storage, demand response, topology optimization, and active management of renewable resources. Such extensions would require tackling a set of different challenges beyond the scope of this letter paper. (iii) Like almost all similar methods, it would be ideal, but difficult and largely open, to provide formal guarantee on DNN's performance after training. It is also an interesting direction to explore unsupervised learning and reinforcement learning for solving single-period and multi-period OPF problems; see [5], [18], [19] for some recent studies along the line with given topology/admittance.

References

[1] 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, 2021.

[2] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *Proc. IEEE SmartGridComm*, 2020.

[3] 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.

[4] 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.

[5] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," in *Proc. ICLR*, 2021.

[6] M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to solve the AC-OPF using sensitivity-informed deep neural networks," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2833–2846, 2021.

[7] M. Chatzos, T. W. Mak, and P. Van Hentenryck, "Spatial network decomposition for fast and scalable AC-OPF learning," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2601–2612, 2021.

[8] X. Pan, M. Chen, T. Zhao, and S. H. Low, "DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems (early access)," *IEEE Syst. J.*, 2022.

[9] Y. Chen, S. Lakshminarayana, C. Maple, and H. V. Poor, "A meta-learning approach to the optimal power flow problem under topology reconfigurations," *arXiv preprint arXiv:2012.11524*, 2020.

[10] Y. Jia, X. Bai, L. Zheng, Z. Weng, and Y. Li, "ConvOPF-DOP: A data-driven method for solving AC-OPF based on CNN considering different operation patterns (early access)," *IEEE Trans. Power Syst.*, 2022.

[11] X. Pan, T. Zhao, and M. Chen, "DeepOPF: Deep neural network for DC optimal power flow," in *Proc. IEEE SmartGridComm*, 2019.

[12] 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, 2020.

[13] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive AC-OPF solutions," *arXiv preprint arXiv:2205.10129*, 2022.

[14] G. Poyrazoglu and H. Oh, "Optimal topology control with physical power flow constraints and N-1 contingency criterion," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3063–3071, 2015.

[15] M. Zhou, "Codes and system files for DeepOPF-FT network for 2022. [Online]. Available:https://github.com/Mzhou-cityu/DeepOPF-FT.

[16] 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, 2010.

[17] S. Babaeinejadsarookolaee, A. B. Birchfield, R. D. Christie, C. Coffrin, C. L. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, C. Josz, R. Korab, B. C. Lesieutre, J. Maeght, D. K. Molzahn, T. J. Overbye, P. Panciatici, B. Park, J. Snodgrass, and R. Zimmerman, "The power grid library for benchmarking AC optimal power flow algorithms," *arXiv preprint arXiv:1908.02788*, 2019.

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

[19] 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.