DeepOPF-AL: Augmented Learning for Solving AC-OPF Problems with a Multi-Valued Load-Solution Mapping

Xiang Pan
Dept. of Information Engineering
The Chinese University of Hong Kong

Wanjun Huang
School of Automation Science and Electrical Engineering
Beihang University

Minghua Chen
School of Data Science
City University of Hong Kong

Steven H. Low
Depts. of CMS and EE
Caltech

ABSTRACT

The existence of multi-valued load-solution mapping in general non-convex problems poses a fundamental challenge to deep neural network (DNN) schemes. A well-trained DNN in the existing supervised learning framework fails to learn the multi-valued mapping accurately and generates inferior solutions. We propose *augmented learning* as a methodological framework to tackle this challenge. We focus on AC-OPF as an important example and develop DeepOPF-AL to solve it. The main idea is to train a DNN to learn a *single-valued* mapping from an augmented input, i.e., (load, initial point), to the solution generated by an iterative OPF solver with the load and initial point as intake. We then apply the learned augmented mapping to solve AC-OPF problems much faster than conventional solvers. Simulation results over IEEE test cases show that DeepOPF-AL achieves noticeably better optimality and similar feasibility and speedup performance as compared to a recent DNN scheme, with the same DNN size yet larger training-data size. We believe the *augmented-learning* approach will find applications in various problems with a multi-valued input-solution mapping.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Neural networks**; • **Hardware** → **Power and energy**.

KEYWORDS

AC optimal power flow; deep neural network; augmented learning

ACM Reference Format:
Xiang Pan, Wanjun Huang, Minghua Chen, and Steven H. Low. 2023. DeepOPF-AL: Augmented Learning for Solving AC-OPF Problems with a Multi-Valued Load-Solution Mapping. In *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3575813.3576874

e-Energy '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0032-3/23/06…$15.00
https://doi.org/10.1145/3575813.3576874

1 INTRODUCTION

Optimal Power Flow (OPF) is a critical but challenging problem in power systems. It aims to find an optimal generation dispatch to meet loads while satisfying non-convex power flow equations and operational constraints [3]. As uncertainty continues to increase with the wider adoption of renewable sources, the need to close the loop on a faster timescale intensifies. Future applications may have to solve OPF faster than traditional methods can deliver. This motivates research leverages supervised-learning based on historical or simulated (load, optimal solution) data. Recent works [9, 20–22, 28] indicate that deep neural networks (DNNs) could obtain desirable solutions in a fraction of the time needed by a conventional solver.

AC-OPF can admit multiple global and local solutions for each load input [17], resulting in a multi-valued load-solution mapping [11]. An increasing number of studies have demonstrated that multiple solutions exist and are hard to exclude even under normal conditions [2, 27]. Remarkably, due to the NP-hardness of AC-OPF problems, it is computationally prohibitive to find all possible solutions and differentiate between them based on the objective cost. Thus, it is essential to consider the information on global and local solutions. The presence of the multi-valued load-solution mapping poses a fundamental challenge to supervised-learning schemes. As illustrated in Fig. 1, a well-trained DNN using the existing supervised-learning scheme [15] fails to learn the target mapping and generates inferior solutions. While recent works attempt to tackle the challenge, the limitation is that they cannot ensure the multi-valued mapping is well-learned. We present detailed discussions of related works in Sec. 2. To date, applying machine learning to solve AC-OPF (as well as the general optimization problems) with multiple solutions for the same input remains largely open.

In this paper, we propose *augmented learning* as a methodological framework to tackle multi-valued-mapping challenge in supervised learning. We focus on AC-OPF and develop DeepOPF-AL to solve it. Our idea is to embed the multi-valued input-solution mapping into a single-valued mapping by augmenting the input with extra parameters. Specifically, DeepOPF-AL trains a DNN to learn the mapping from (load, initial point) to the unique OPF solution generated by the primal-dual interior-point method with the load and initial point as intake. Simulation results on IEEE test cases show that DeepOPF-AL achieves better AC-OPF optimality and similar feasibility and speedup performance, as compared to a recent scheme [14], with the same DNN size yet larger training-data size. The results also verify the robust performance of DeepOPF-AL.

2 RELATED WORK

Machine learning has been employed for solving optimization problems with hard constraints [8, 31] like AC-OPF. Some works utilize DNN to facilitate conventional methods by identifying active constraints upon a load input [5, 6, 18, 23, 29] or directly predict solutions by learning the load-solution mapping [4, 7, 9, 12, 16, 19, 21, 22, 24, 26, 28, 30, 31]. Nevertheless, existing DNN-based methods explicitly or implicitly assume the target mapping is single-valued. By the universal approximation theorem, NN could well approximate such a mapping. However, the obliviousness to multi-valued mapping could lead to inferior performances of existing supervised-learning schemes. See an illustrating example in Sec. 3.

Recently, there have been several works to tackle the challenge of learning a multi-valued mapping. The method in [15] properly selects one solution for each load input as the ground truth by solving a challenging bi-level optimization problem. However, the exact optimal solution is computationally prohibitive due to the NP-hardness, and thus there is no guarantee to learn the multi-valued load-solution mapping. The other is applying unsupervised learning to train DNN without ground-truths [8, 13]. However, without ground truths, it cannot preserve the target load-solution relationship, and the trained DNN could generate inferior solutions.

In this paper, we propose *augmented learning* framework as a simple but effective methodology to tackle the multi-valued input-solution mapping challenge. We develop DeepOPF-AL to learn a unique mapping from an *augmented* load input to the corresponding solution from a deterministic iterative solver for solving AC-OPF. Note that DeepOPF-AL is different from the approach in [1] in that it directly outputs the solution in one pass, while the latter is an iterative scheme that replaces the update function in Newton's method with a DNN. DeepOPF-AL is similar in spirit to the method in an independent work [25] on learning the multi-valued input-solution mapping. The method in [25] develops a DNN scheme for solving power control problems in wireless communication by learning the input-solution mapping generated by a deterministic iterative solver using a *fixed* initial point for all inputs. DeepOPF-AL generalizes the method in [25] by learning the mapping from (load, initial point) to the solution generated by a deterministic iterative solver by augmenting the load with *changeable* initial points that could preserve the multi-valued load-solution relationship.

3 REVIEW OF AC-OPF

The standard AC-OPF problem is formulated as

$$\min \sum_{i\in\mathcal{N}} C_i\left(P_{Gi}\right) \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in\mathcal{E}} \mathrm{Re}\left\{V_i\left(V_i^*-V_j^*\right)y_{ij}^*\right\} = P_{Gi}-P_{Di}, i\in\mathcal{N}, \tag{2}$$

$$\sum_{(i,j)\in\mathcal{E}} \mathrm{Im}\left\{V_i\left(V_i^*-V_j^*\right)y_{ij}^*\right\} = Q_{Gi}-Q_{Di}, i\in\mathcal{N}, \tag{3}$$

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max}, i\in\mathcal{N}, \tag{4}$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}, i\in\mathcal{N}, \tag{5}$$

$$V_i^{\min} \le |V_i| \le V_i^{\max}, i\in\mathcal{N}, \tag{6}$$

$$|V_i\left(V_i^*-V_j^*\right)y_{ij}^*| \le S_{ij}^{\max}, (i,j)\in\mathcal{E}, \tag{7}$$

$$\text{var. } P_{Gi}, Q_{Gi}, V_i, i\in\mathcal{N}.$$

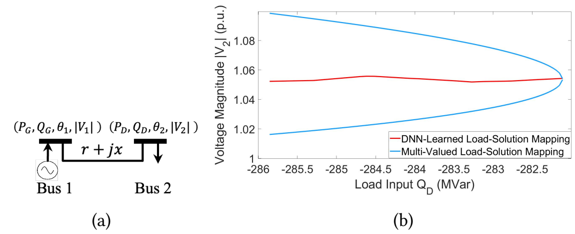


Figure 1: (a) A 2-bus power network with $|V_1| = 0.9$ p.u., $\theta_1 = 0$, $r = 0$ p.u., $x = 0.25$ p.u., and $P_D = 343$ MW. (b) The AC-OPF problem in (1)-(6) over the 2-bus network has a multi-valued mapping from the load Q_D to $|V_2^*|$ in the optimal solution. A well-trained DNN with two hidden-layer (300 neurons on each hidden layer), and with 100 uniformly-sampled $(Q_D, |V_2^*|)$ pairs and ℓ_2 distance as the loss function fails to learn the target mapping. The average relative difference between the ground-truth and prediction is 3%.

$\mathrm{Re}\{z\}$, $\mathrm{Im}\{z\}$, z^*, and $|z|$ denote the real part, the imaginary part, the conjugate, and the magnitude of a complex variable z, respectively. $\mathcal{N}$ and $\mathcal{E}$ denote the set of buses and the set of branches, respectively. P_{Gi} (resp. Q_{Gi}) and P_{Di} (resp. Q_{Di}) denote the active (resp. reactive) power generation and active (resp. reactive) load on bus i, respectively. V_i represents complex voltage, including the magnitude $|V_i|$ and the phase angle θ_i, on bus i. y_{ij} and $S_{ij}^{\max}$ denote the admittance and the branch flow limit of the branch $(i,j)\in\mathcal{E}$, respectively. Equations (2) and (3) represent power-flow balance equations. Constraints (4) and (5) represent the active and reactive generation limits. Constraints (6) and (7) represent the voltage magnitude and the branch flows limits. The objective is to minimize the total cost of active power generation, where $C_i(\cdot)$ is the quadratic cost function of the generator at bus i. We set $C_i(P_{Gi}) = 0$ and $P_{Gi}^{\min} = P_{Gi}^{\max} = Q_{Gi}^{\min} = Q_{Gi}^{\max} = 0$ if bus i has no generator.

As mentioned in Sec. 1, AC-OPF problems may admit multiple (global and local) optimal solutions for a load input, resulting in multi-valued load-solution mapping. Let us consider the concrete 2-bus system shown in Fig. 1. Specifically, the power network contains a generator at bus-1 and a load input at bus-2. Suppose we fix $|V_1| = 0.9$ p.u., $\theta_1 = 0$, $r = 0$ p.u., $x = 0.25$ p.u., and $P_D = 343$ MW. We then depict the relationship between the Q_D and the $|V_2|$ by solving the AC-OPF problem in (1)-(6) with Q_D as the varying load input. The multi-valued load-solution (Q_D to $|V_2^*|$) mapping is illustrated on Fig. 1. As observed, the existing supervised-learning approach fails to learn the target mapping. The reason is that the uniformly-sampled "mixed" data pairs (i.e., the data come from both upper and lower parts of the mapping (in blue) for different load inputs at the same time), making trained DNN generate solutions that lie in the middle to achieve minimum ℓ_2 distance.

4 DEEPOPF-AL: AUGMENTED LEARNING FOR SOLVING AC-OPF

The schematic of the developed DeepOPF-AL is shown in Fig. 2. Following a particular augmented-learning design, DeepOPF-AL trains a DNN to learn the mapping from (load, initial point) to the unique AC-OPF solution generated by the primal-dual interior

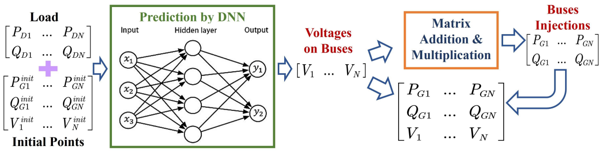
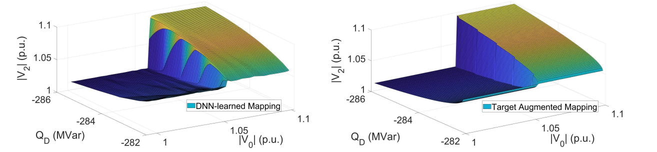


Figure 2: Illustration of the DeepOPF-AL approach.



(a) (b)

Figure 3: Comparisons of the target augmented mapping and DNN-learned mapping for the 2-bus case. ($|V_0|$ is the initial point of $|V_2|$). (a) Target augmented mapping. (b) DNN-learned augmented mapping. The average relative prediction error is 0.05%, which is smaller than 3% as reported in Fig. 1.

point method with the load and initial point as intake. The following proposition inspires our design of DeepOPF-AL.

Proposition 1. *Let $\mathbf{X}^*$ be the solution generated by a deterministic iterative solver with the given initial point $\mathbf{X}_0$ and load input $\mathbf{D}$. Then, the mapping from $(\mathbf{X}_0, \mathbf{D})$ to the corresponding final solution, described as $\mathbf{X}^*(\mathbf{X}_0, \mathbf{D}) = \psi^*(\mathbf{X}_0, \mathbf{D})$ is single-valued.*

Proposition 1 indicates the augmented mapping fully incorporates the multi-valued relationship between the load input and generated solutions. This observation motivates us to train a DNN to approximate the augmented mapping $\psi^*(\cdot)$. Such a design avoids obtaining training data with a mixture of multiple solutions and preserves the multi-valued load-solution relationship.

As shown in Fig. 2, DeepOPF-AL trains a DNN to predict the bus voltages and reconstructs the bus injections, i.e., RHS values of (2)-(3), and finally the generations, all by simple scalar calculation. Such a predict-and-reconstruct framework [14, 21, 22] guarantees the power-flow equality constraints and reduces the number of variables to predict. Lastly, DeepOPF-AL employs the post-processing process in [14] to help keep the obtained solution within the box constraints in (4)-(6). Existing works [14, 20, 28] have demonstrated the feed-forward network's desirable performance in solving AC-OPF problems due to its universal mapping approximation capability, simplicity and scalability. Thus, we apply the multi-layer feed-forward neural network structure to build the DNN model. We refer to Appendices A and B for details. We also remark that augmented learning is a methodological framework that does not specify NN structures or training processes. Therefore, any other NN structures or training process could be used in DeepOPF-AL.

4.1 Discussions

Solving AC-OPF problems with a multi-valued load-solution mapping is challenging for DNN schemes. As compared to the existing method [15], DeepOPF-AL guarantees that the DNN learns a single-valued mapping by using a simple uniformly-sample strategy without the need to solve the bi-level problem. In application, for each test load input, one can run DeepOPF-AL several times with different initial points and output the least-cost solution, without the need to differentiate between global or local solutions. [1]

DeepOPF-AL also excels in the following aspects: (i) It can learn the augmented mapping of any deterministic iterative algorithm. We choose to learn that of the popular primal-dual interior point method, which is observed to return good-quality OPF solutions in practice. (ii) For the set of inputs (load, initial point) for which the primal-dual interior point method fails to converge, DeepOPF-AL can still generate solutions with decent optimality performance, which is shown in Sec. 5. This indicates better practicability of DNN schemes over iterative solvers, in addition to speedup. Nevertheless, DeepOPF-AL learns a single-valued higher-dimensional augmented mapping but requires a larger DNN and more training data than learning a single-valued load-solution mapping.

[1] One could also output the OPF solution by other criteria, like the stability of the solution, the distance from the grid's current state, and robustness of the solution.

5 NUMERICAL EXPERIMENTS

5.1 Setup

We conduct simulations in CentOS 7.6 with a quad-core (i7-3770@3.40G Hz) CPU and 16GB RAM. We compare the performance of DeepOPF-AL and a recent scheme DeepOPF-V [14] that first learns the mapping between loads and voltages of all buses, and then computes the bus generations, on the 2-bus (as illustrated in Fig. 3) and IEEE 39-/300-bus systems. Three test cases correspond to three representative scenarios: the 2-bus system has a multi-valued load-solution mapping, the modified IEEE 39-bus case has a multi-valued load-solution mapping, and the IEEE 300-bus case has a single-valued load-solution mapping (a special case of multi-valued mapping). For detailed training setting, please refer to Appendix B. In the simulation, for each test load, we run DeepOPF-AL with randomly-sampled initial points to generate solutions and output the least-cost one. We evaluate the performance with the following metrics: (i) **Optimality loss**: η_{opt} measures the relative optimality difference between the objective values obtained by DeepOPF-AL and MIPS. Closer to zero is better. (ii) **Running time and speedup**: t_{mips} and t_{dnn} represent the average running times of the MIPS solver and the DNN schemes. η_{speed} measures the corresponding average speedup ratios. Higher is better. (iii) **Constraint satisfaction**: η_{P_G}, η_{Q_G}, and η_{S_l} measure the average constraint satisfaction percentages for active/reactive generation and branch flow limit, respectively. Higher is better. (iv) **Load satisfaction**: η_{P_D} and η_{Q_D} measure the average load-serving mismatch percentage of active and reactive loads, respectively. Closer to 0 is better.

5.2 Performance Evaluation

5.2.1 2-bus system. We visualize the target augmented mapping and DNN-learned augmented mapping for the 2-bus example in Fig. 3, respectively. Fig. 3 verifies that DeepOPF-AL ensures that the DNN learns the single-valued augmented mapping well. Meanwhile, the visualization (as shown in Fig 3(b) demonstrates the significant improvements in the solution quality as compared to the existing DNN schemes (as shown in Fig. 1).

Table 1: Simulation results for the modified IEEE 39-bus system.

Metric	Case39-V1 with Avg. Cost Diff. = 0.36%				Case39-V2 with Avg. Cost Diff. = 30%			
	Balanced Dataset		Unbalanced Dataset		Balanced Dataset		Unbalanced Dataset	
	DeepOPF-AL	DeepOPF-V	DeepOPF-AL	DeepOPF-V	DeepOPF-AL	DeepOPF-V	DeepOPF-AL	DeepOPF-V
η_{opt}(%)	**−0.77**	1.33	**−0.41**	0.31	**0.48**	-8.56	**0.66**	-5.98
η_{P_G}(%)/η_{Q_G}(%)	97.61/**92.04**	99.96/91.68	97.54/**94.26**	99.99/90.73	97.5/**94.6**	99.3/91.7	97.4/**98.4**	99.8/96.7
η_{S_l}(%)	100	100	100	100	100	100	100	100
η_{P_D}(%)/η_{Q_D}(%)	**−0.42/8.06**	1.41/19.65	**−0.32/2.61**	0.39/8.25	**0.19/6.47**	0.61/27.6	**0.09/0.49**	0.32/12.9
t_{mips} (ms)	3366	3366	3317	3317	2808	2808	2676	2676
t_{dnn} (ms)	1.2	1.1	1.2	1.1	1.4	1.3	1.3	1.2
η_{speed}	×2805	×3060	×2764	×3015	×2006	×2160	×2058	×2230

Table 2: Simulation results for the IEEE 300-bus system.

Metric	Convergent Dataset		Non-convergent Dataset	
	DeepOPF-AL	DeepOPF-V	DeepOPF-AL	MIPS
η_{opt}(%)	0.01	-0.01	**−0.13**	20.4
η_{P_G}(%)	100	100	**99.9**	23.2
η_{Q_G}(%)	100	100	**100**	78.3
η_{S_l}(%)	100	100	**100**	80.7
η_{P_D}(%)	0.0	-0.01	**−0.1**	-73.4
η_{Q_D}(%)	0.02	0.02	**−0.02**	-109.2
t_{mips}(ms)	4245	4245	-	-
t_{dnn}(ms)	2.2	1.8	2.2	-
η_{speed}	×1930	×2358	-	-

5.2.2 IEEE 39-bus system. We evaluate all DNN approaches on two modified IEEE 39-bus systems, where the load inputs in Case39-V1 and Case39-V2 have two solutions with on average 0.36% and 30% difference in objective values. The results are shown in Table 1, and we have the following observations. First, DeepOPF-AL achieves better performance in reactive generation constraint satisfaction, reactive load satisfaction and optimality loss than DeepOPF-V (marked in bold in Table 1), while getting similar performance in other metrics in both datasets. Second, DeepOPF-AL performs more consistently than DeepOPF-V over the two datasets on each test case. It demonstrates the effectiveness of our augmented-learning approach. Third, the improvement of DeepOPF-AL on the performance metrics is more significant for the case where a load input has two solutions with larger cost differences. The empirical explanation for this observation may be that larger cost differences correspond to more distinct solutions (i.e., solutions with longer distances to each other). Consequently, DeepOPF-AL achieves better performance, than DeepOPF-V.

5.2.3 IEEE 300-bus system. The results for the IEEE 300-bus system in Table 2 show that DeepOPF-AL achieves a similar optimality gap, feasibility, and speed-up performance, as compared to DeepOPF-V. These results confirm that, for this setting with a single-valued load-solution mapping, the (higher-dimensional) augmented mapping to learn by DeepOPF-AL is degenerated and can be represented using the same-size DNN as the (lower-dimensional) load-solution one to learn by DeepOPF-V. Meanwhile, the augmented mapping still requires more training data if one has no prior knowledge of its degenerated dimension, as in this simulation.

Overall, the simulation results on the 2-bus, modified IEEE 39-bus, and IEEE 300-bus systems show the promising performance of DeepOPF-AL. The results indicate that DeepOPF-AL can achieve desirable performance, regardless of whether or not the AC-OPF problems have a multi-valued load-solution mapping. As compared, the existing DNN scheme (i.e.,DeepOPF-V) only works well in the case with a single-valued load-solution mapping. Meanwhile, as discussed in Sec. 1, recent work [15] requires solving a bi-level in data generation procedure to learn the multi-valued load-solution mapping, in which the optimal solution is computationally prohibitive. In addition, the trained DNN fails to learn the target multi-valued mapping accurately and generates inferior solutions. In contrast, DeepOPF-AL trains DNN with a simple uniformly-sampling data-generation procedure without the need to solve the challenging bi-level problem. We remark that it is difficult to determine the type (single-valued or multi-valued) of load-solution mapping. Thus, DeepOPF-AL is more applicable in practice due to its convenience in data generation and no requirement for prior knowledge of the target multi-valued mapping.

5.3 Robustness of DeepOPF-AL

We test the robustness of DeepOPF-AL on IEEE 300-bus using the test load inputs, where we randomly sample initial points for each load for which the MIPS solver fails to converge. We observe that DeepOPF-AL still attains decent performance (marked in bold in Table 2), while the non-convergent solutions by the MIPS solver suffer from significant performance degradation. The explanation for this observation is that DeepOPF-AL could directly generate solutions upon the given load input and initial points, regardless of whether or not MIPS convergences with the given input and is likely to generalize well to the unseen input region. This implies that DeepOPF-AL, while trained with data generated by the MIPS solver, achieves a more robust performance than the MIPS solver.

6 CONCLUDING REMARK

We propose augmented learning as a methodological framework to solve the general optimization problems with multi-valued mappings. We develop DeepOPF-AL as the first augmented-learning-based approach and apply it to solve AC-OPF problems with a multi-valued mapping. Simulation results show that it achieves a better optimality gap ($<0.7\%$) and similar feasibility and speedup (up to *three* orders of magnitude faster than the conventional solver) performances compared to a recent DNN scheme, yet with a larger training-data size. A future direction is to improve augmented-learning designs for better training efficiency and extend it to other problems in supervised learning with multi-valued input-solution mappings. It is also an interesting direction to extend the augmented learning approach to other DNN architectures (e.g., convolutional neural networks and graph neural networks).

REFERENCES

[1] Kyri Baker. 2020. A Learning-boosted Quasi-Newton Method for AC Optimal Power Flow. https://doi.org/10.48550/ARXIV.2007.06074

[2] Waqquas A Bukhsh, Andreas Grothey, Ken IM McKinnon, and Paul A Trodden. 2013. Local solutions of the optimal power flow problem. *IEEE Transactions on Power Systems* 28, 4 (2013), 4780–4788.

[3] Mary B Cain, Richard P O'neill, and Anya Castillo. 2012. History of optimal power flow and formulations. *Federal Energy Regulatory Commission* 1 (2012), 1–36.

[4] Minas Chatzos, Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. 2020. High-Fidelity Machine Learning Approximations of Large-Scale Optimal Power Flow. https://doi.org/10.48550/ARXIV.2006.16356

[5] Yize Chen and Baosen Zhang. 2020. Learning to Solve Network Flow Problems via Neural Decoding. https://doi.org/10.48550/ARXIV.2002.04091

[6] Deepjyoti Deka and Sidhant Misra. 2019. Learning for DC-OPF: Classifying active sets using neural nets. In *Proc. IEEE Milan PowerTech*. IEEE.

[7] Roel Dobbe, Oscar Sondermeijer, David Fridovich-Keil, Daniel Arnold, Duncan Callaway, and Claire Tomlin. 2019. Towards distributed energy services: Decentralizing optimal power flow with machine learning. *IEEE Trans. Smart Grid* 11 (2019), 1296–1306.

[8] Priya L Donti, David Rolnick, and J Zico Kolter. 2021. DC3: A learning method for optimization with hard constraints. In *Proc. ICLR*.

[9] Ferdinando Fioretto, Terrence WK Mak, and Pascal Van Hentenryck. 2020. Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods. In *Proc. AAAI*, Vol. 34. AAAI Press, 630–637.

[10] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep Learning*. Vol. 1. MIT Press Cambridge.

[11] Lech Górniewicz and Lech Górniewicz. 2006. *Topological Fixed Point Theory of Multivalued Mappings*. Vol. 4. Springer.

[12] Neel Guha, Zhecheng Wang, Matt Wytock, and Arun Majumdar. 2019. Machine Learning for AC Optimal Power Flow. https://doi.org/10.48550/ARXIV.1910.08842

[13] Wanjun Huang and Minghua Chen. 2021. DeepOPF-NGT: A Fast Unsupervised Learning Approach for Solving AC-OPF Problems without Ground Truth. In *ICML Workshop on Tackling Climate Change with Learning*.

[14] Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H Low. 2022. DeepOPF-V: Solving AC-OPF Problems Efficiently. *IEEE Trans. Power Syst.* 37, 1 (2022), 800–803.

[15] James Kotary, Ferdinando Fioretto, and Pascal Van Hentenryck. 2021. Learning Hard Optimization Problems: A Data Generation Perspective. https://doi.org/10.48550/ARXIV.2106.02601

[16] Xingyu Lei, Zhifang Yang, Juan Yu, Junbo Zhao, Qian Gao, and Hongxin Yu. 2021. Data-Driven Optimal Power Flow: A Physics-Informed Machine Learning Approach. *IEEE Trans. Power Syst.* 36, 1 (Jan. 2021), 346–354.

[17] Daniel K Molzahn, Ian A Hiskens, et al. 2019. A Survey of Relaxations and Approximations of the Power Flow Equations. *Foundations and Trends® in Electric Energy Systems* 4, 1-2 (2019), 1–221.

[18] Yeesian Ng, Sidhant Misra, Line A Roald, and Scott Backhaus. 2018. Statistical learning for DC optimal power flow. In *PSCC*. 1–7.

[19] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. 2020. Optimal power flow using graph neural networks. In *Proc. IEEE ICASSP*. IEEE, Barcelona, Spain, 5930–5934.

[20] Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H. Low. 2022. DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems. *IEEE Systems Journal* (2022), 1–11.

[21] Xiang Pan, Tianyu Zhao, and Minghua Chen. 2019. DeepOPF: Deep Neural Network for DC Optimal Power Flow. In *Proc. IEEE SmartGridComm*. Beijing, China. available on arXiv in May 2019 (arXiv:1905.04479v1).

[22] Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. 2021. DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow. *IEEE Trans. Power Syst.* 36, 3 (May 2021), 1725–1735.

[23] S. Pineda, J. M. Morales, and A. Jiménez-Cordero. 2020. Data-Driven Screening of Network Constraints for Unit Commitment. *IEEE Trans. Power Syst.* 35, 5 (2020), 3695–3705.

[24] Manish K. Singh, Vassilis Kekatos, and Georgios B. Giannakis. 2021. Learning to Solve the AC-OPF using Sensitivity-Informed Deep Neural Networks. https://doi.org/10.48550/ARXIV.2103.14779

[25] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D Sidiropoulos. 2018. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing* 66, 20 (2018), 5438–5453.

[26] A. Velloso and P. Van Hentenryck. 2021. Combining Deep Learning and Optimization for Preventive Security-Constrained DC Optimal Power Flow. *IEEE Trans. Power Syst., to be published* 36, 4 (2021), 3618–3628. https://doi.org/10.1109/TPWRS.2021.3054341

[27] Dan Wu, Daniel K Molzahn, Bernard C Lesieutre, and Krishnamurthy Dvijotham. 2017. A deterministic method to identify multiple local extrema for the ac optimal power flow problem. *IEEE Transactions on Power Systems* 33, 1 (2017), 654–668.

[28] Ahmed S. Zamzam and Kyri Baker. 2020. Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 1–6. https://doi.org/10.1109/SmartGridComm47815.2020.9303008

[29] Ling Zhang, Yize Chen, and Baosen Zhang. 2020. A Convex Neural Network Solver for DCOPF with Generalization Guarantees. https://doi.org/10.48550/ARXIV.2009.09109

[30] Tianyu Zhao, Xiang Pan, Minghua Chen, and Steven H. Low. 2021. Ensuring DNN Solution Feasibility for Optimization Problems with Convex Constraints and Its Application to DC Optimal Power Flow Problems. https://doi.org/10.48550/ARXIV.2112.08091

[31] Tianyu Zhao, Xiang Pan, Minghua Chen, Andreas Venzke, and Steven H. Low. 2020. DeepOPF+: A Deep Neural Network Approach for DC Optimal Power Flow for Ensuring Feasibility. In *Proc. IEEE SmartGridComm*. IEEE, Tempe, AZ, USA, 1–6.

[32] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, Robert John Thomas, et al. 2011. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.* 26, 1 (Feb. 2011), 12–19.

A DNN ARCHITECTURE AND TRAINING

Recall that DeepOPF-AL adopts the prediction-and-reconstruction framework [14]. It trains DNN to predict the bus voltages and reconstructs the bus injections, i.e., RHS values of (2)-(3), and finally the generations, all by simple scalar calculation. Such a predict-and-reconstruct framework [14, 21, 22] guarantees the power flow equality constraints and reduces the number of variables to predict. Lastly, DeepOPF-AL employs the post-processing process in [14] to help keep the obtained solution within the box constraints in (4)-(6). To enforce the feasibility of predicted variables, we associate with the predicted variable a one-to-one corresponding scaling factor [21] by

$$x_{dnn} = s_{dnn} \cdot \left(x^{\max} - x^{\min}\right) + x^{\min}, \tag{8}$$

where s_{dnn} is the scaling factor. The DNN predicts s_{dnn} and compute x_{dnn} by (8). We apply the Sigmoid function [10] as the activation function of the output layer to ensure the predicted scaling factor within $(0, 1)$. We build a DNN model using the multi-layer feed-forward neural network structure:

$$\mathbf{s}_0 = [\mathbf{D}, \mathbf{X}_0], \tag{9}$$

$$\mathbf{s}_i = \sigma\left(W_i \mathbf{s}_{i-1} + \mathbf{b}_i\right), \forall\, i = 1, \ldots, L, \tag{10}$$

$$\mathbf{s}_{dnn} = \sigma'\left(W_{L+1} \mathbf{s}_L + \mathbf{b}_{L+1}\right). \tag{11}$$

In our design, the input load ($\mathbf{D}$) and initial points for the primal variables ($\mathbf{X}_0$) forms DNN's input $\mathbf{s}_0$. $\mathbf{s}_i$ is the output vector of the i-th hidden layer, depending on weights W_i, biases $\mathbf{b}_i$ and the $(i-1)$-th layer's output $\mathbf{s}_{i-1}$. W_i and $\mathbf{b}_i$ are adjustable DNN's parameters. L is the number of hidden layers. $\sigma(\cdot)$ and $\sigma'(\cdot)$ are ReLU and Sigmoid activation functions, respectively.

During training, we design the loss function for each instance as the mean square error between the generated solution and the ground truth (the generated AC-OPF solution). The training process aims to minimize the average loss of the training dataset by tuning the DNN's parameters W_i and $\mathbf{b}_i$:

$$\min_{W_i, \mathbf{b}_i, i=1,\ldots,L} \frac{1}{\mathrm{card}\left(\mathcal{T}\right)} \sum_{k \in \mathcal{T}} \left\|\mathbf{s}_{dnn,k} - \mathbf{s}_{gt,k}\right\|_2^2, \tag{12}$$

where $\mathcal{T}$ is the training data-set, $\mathbf{s}_{dnn,k}$ and $\mathbf{s}_{gt,k}$ are the prediction and ground-truth for the k-th instance, respectively. We apply the Adam algorithm to update the DNN's parameters.

B DETAILS OF TRAINING SETTINGS FOR DEEPOPF-AL ON DIFFERENT TEST SYSTEMS

B.1 2-bus system

We randomly generated 100 initial points for each input (100 load inputs in total) and feed them with the loads into the Matpower Interior Point Solver (MIPS) solver [32], which implements the primal-dual interior point method, to obtain reference AC-OPF solutions. We train the DNN with the same hidden layers and widths given in Sec. 3, i.e., 2 hidden layers with 300 neurons on each hidden layer. We set the batch size, maximum epoch, and learning rate to be 64, 4000, and 1e-4, respectively.

B.2 39-bus system

We adopt two modified IEEE 39-bus systems [2]:

- Case39-V1: The modifications include: (i) scaling the active/reactive power generation bounds by 4, applying ±5% voltage bounds and linear objective coefficients.
- Case39-V2: The modification applies ±5% voltage bounds.

For the evaluation of IEEE 39-bus systems, we generate a realistic load on each bus by multiplying the default value by an interpolated demand curve based on 11-hour California's net load in Jul. – Sept. 2021, with a time granularity of 30 seconds, thus 2,760 load instances per day. For each load, we randomly generate initial points and feed them together with the loads into the MIPS solver [32] to obtain reference AC-OPF solutions. For each test case, we design two datasets: a balanced dataset (with 75,562 and 89,972 data points for Case39-V1 and Case39-V2, respectively; for each load, the ratio between the numbers of the low-cost solutions and the high-cost solutions is 1:1) and an unbalanced dataset (with 41,123 and 52,384 data points for Case39-V1 and Case39-V2, respectively; for each load, the ratio between the numbers of the low-cost solutions and the high-cost solutions are 9:1). We split each dataset using the "80/20" strategy to obtain the training and test sets. We build DNN models consisting of 3 hidden layers with 1024/768/512 neurons for the modified 39-bus system. We set the batch size, maximum epoch, and learning rate to 128, 4000, and 1e-4, respectively.

B.3 300-bus system

For the evaluation of IEEE 300-bus systems, we generate a realistic load in the same way as the 39-bus system. The training set for DeepOPF-V contains 2,760 (load, solution) data pairs. The training set for DeepOPF-AL has 110,400 ((load, initial point), solution) data points, where we randomly sample 40 initial points for each load. We build DNN models consisting of 4 hidden layers with 1024/768/512/256 neurons for the 300-bus system. We set the batch size, maximum epoch, and learning rate to 128, 4000, and 1e-4, respectively. We evaluate the performance of DeepOPF-AL and DeepOPF-V using the same-size DNN and same hyper-parameters, following the "80/20" training/testing splitting rule.

B.4 Robustness of DeepOPF-AL

We test the robustness of DeepOPF-AL on IEEE 300-bus using the 2760 test load inputs. Specifically, the test set has 55,200 (load, initial point) data points, where we randomly sample 20 initial points for each load for which the MIPS solver fails to converge.