Revisiting Deep AC-OPF

Oluwatomisin I. Dada
University of Cambridge
oid20@cam.ac.uk

Neil D. Lawrence
University of Cambridge
ndl21@cam.ac.uk

Abstract

Recent work has proposed machine learning (ML) approaches as fast surrogates for solving AC optimal power flow (AC-OPF), with claims of significant speed-ups and high accuracy. In this paper, we revisit these claims through a systematic evaluation of ML models against a set of simple yet carefully designed linear baselines. We introduce **OPFormer-V**, a transformer-based model for predicting bus voltages, and compare it to both the state-of-the-art DeepOPF-V model and simple linear methods. Our findings reveal that, while OPFormer-V improves over DeepOPF-V, the relative gains of the ML approaches considered are less pronounced than expected. Simple linear baselines can achieve comparable performance. These results highlight the importance of including strong linear baselines in future evaluations.

1 Introduction

Reliable electricity distribution is a vital part of modern society. The growing efforts to decarbonize, driven by climate change, demand greater electrification and integration of renewable sources such as wind and solar. However, variability in renewable generation and dynamic demand increase uncertainty, requiring grid operators to solve AC optimal power flow (AC-OPF) problems more frequently. This has motivated extensive research on machine learning (ML) approaches for OPF [Donti and Kolter, 2021, Donti et al., 2020, Huang et al., 2021, Donon et al., 2020, Owerko et al., 2022].

ML-based OPF methods can be broadly categorized into *direct* and *hybrid* approaches [Falconer and Mones, 2022]. *Direct* methods learn a mapping from grid parameters to OPF solutions, offering substantial speedups [Falconer and Mones, 2020, Owerko et al., 2020, Hansen et al., 2023, Donti et al., 2021, Liu et al., 2022, Huang et al., 2021]. In contrast, *hybrid* methods leverage predictions, e.g. warm starts, dual variables, or reduced formulations, to accelerate conventional solvers [Robson et al., 2019, Pham and Li, 2022, Falconer and Mones, 2022].

We focus on DeepOPF-V, a state-of-the-art *direct* method by Huang et al. [2021]. Unlike previous works, DeepOPF-V predicts bus voltage magnitudes (V_m) and angles (V_a), from which generator outputs (S_g) are computed using the grid admittance (Y_{bus}) and load (S_l). This formulation enforces voltage bounds directly while other constraints are satisfied to the extent of prediction accuracy, yielding strong empirical performance with high feasibility, low optimality gaps, and large speedups.

In this work, we introduce a transformer-based model, OPFormer-V, which predicts the same output. OPFormer-V consistently outperforms DeepOPF-V across datasets, and we compare both ML models against simpler baselines, finding that linear models achieve comparable performance.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

2 Background

2.1 Optimal power flow (OPF)

OPF is a constrained optimization problem that minimizes the generation cost subject to physical and operational constraints. It can be viewed as an operator $\Phi(\cdot)$ mapping loads $\mathbf{S_l}$, admittance matrix $\mathbf{Y}$, objective $f(\cdot)$ and constraints $\mathcal{C}^E, \mathcal{C}^I$ to generator setpoints $\mathbf{S_g}$. Numerous OPF variants exist depending on additional constraints; here, we focus on the economic dispatch formulation.

Concisely, OPF can be expressed as

$$\begin{aligned} &\min_{\mathbf{z}} f(\mathbf{x}, \mathbf{z}) \\ \text{s.t. } &c_j^E(\mathbf{x}, \mathbf{z}) = 0 \quad j = 1 \dots m \\ &c_k^I(\mathbf{x}, \mathbf{z}) \geq 0 \quad k = 1 \dots n, \end{aligned} \tag{1}$$

where $\mathbf{x}$ denotes the grid parameters and $\mathbf{z}$ optimization variables. $\mathcal{C}^E$ represents the set of equality constraints given by the power flow equations and $\mathcal{C}^I$ represents the set of inequality constraints on voltage magnitudes, thermal branch limits and generator ouptuts.

3 Related works

3.1 Direct & hybrid approaches

Early direct methods used GNNs to predict generation setpoints either in supervised [Owerko et al., 2020] or unsupervised [Owerko et al., 2022] settings. Other works exploit graph duals [Hansen et al., 2023], compare architectures [Falconer and Mones, 2020], or enforce feasibility with physics-aware regularization [Liu et al., 2022]. DC3 [Donti et al., 2021] generalizes this idea by embedding feasibility guarantees via differentiable procedures. DeepOPF-V [Huang et al., 2021] advanced the field by predicting bus voltages, from which generator outputs are derived. Zhou et al. [2023] and Liang and Zhao [2023] extend this work by employing the DeepOPF-V framework but training a single model across flexible topologies and various grids respectively.

Hybrid approaches warm-start or simplify solvers by predicting primal/dual variables or non-binding constraints. This ensures feasibility but is slower than direct methods. Examples include meta-optimization for reduced OPF [Robson et al., 2019], GNN-based reduction [Pham and Li, 2022], constraint prediction [Falconer and Mones, 2022], and solver emulation [Baker, 2022, Piloto et al., 2024].

3.2 Linear power flow

The main source of AC-OPF non-convexity lies in the power flow equations. Linear approximations, such as DC-OPF, are widely adopted for convex formulations. Variants incorporate reactive power [Zhang et al., 2013], logarithmic voltage transforms [Li et al., 2018], or squared voltage variables [Li et al., 2018]. A comparative study by Li et al. [2022] concluded that DC-OPF achieved the best overall performance.

4 Datasets

For this work, we evaluated our methods on 2 synthetic self-generated datasets. These datasets are based on the IEEE case 30 and IEEE case 118 grids, having 30 and 118 buses/nodes respectively and generated following the common approach of taking the nominal load case of the format and sampling random variations around that nominal loading scenario. We consider a $\pm 50\%$ variation from the nominal load case at each node and perform latin hypercube sampling. This generates a loading scenario that is then solved using MATPOWER [Zimmerman et al., 1997] and only scenarios that converge to a solution are included in the dataset. The 30 bus and 118 bus datasets contain 100k samples each. For all datasets, we employ a 60/15/25 train/val/test data split.

5 Methods

5.1 Baselines

We consider three simple baseline predictors. **Gridwise averaging** uses the mean v^{train} across all nodes and samples, serving as a data-driven flat start; while it performs the worst for voltage angles (since it assumes no power flow), it predicts magnitudes better than DC-OPF's magnitude assumption of 1.0pu. **Nodewise averaging** instead averages v^{train} per node, producing fixed flows between nodes and achieving regression performance comparable to DeepOPF-V (Figure 1, Tables 3, 4). Finally, **linear regression** trains $2N$ ordinary least squares models, one per node and variable, mapping $\mathbf{S_l}$ to voltage magnitudes and angles.

5.2 Linear power flow

We study two linearized OPF variants that combine approximate power flow equations with a conventional optimizer, ensuring that generator outputs and voltages remain within their bounds. However, these values are no longer coupled via the actual power flow equations, so in order to test the feasibility of the solution, we select a subset of these variables and solve for the remaining variables using the power flow equations. **DC-OPF** employs standard assumptions (unit voltage magnitudes, small-angle approximation, and zero line resistance) equivalent to a first-order Taylor expansion at the flat start, yielding active power flows dependent only on angle differences. In contrast, the **hot start** approach linearizes around nodewise average voltages, solved via CVXPY [Agrawal et al., 2018, Diamond and Boyd, 2016], and models both active and reactive flows including losses as shown in equations 2 - 5. The chosen reference minimizes expected truncation error, with bounds derived from second-order moments of voltage magnitudes and angle differences as shown in equation 7. For efficiency, we approximate the average nodal voltage over multiple loading scenarios as the nodal voltage for the average loading scenario, an assumption justified by the bounded curvature of the load–voltage mapping and the low variance in practical datasets as shown in equation 13.

5.3 ML approaches

DeepOPF-V [Huang et al., 2021] is an FCNN trained with an L2 loss to jointly predict bus voltage magnitudes and angles. Although the original work includes a post-processing step to reduce generator limit violations, from the reported results its effect was minimal, and so we do not implement this post-processing for both ML approaches. DeepOPF-V achieved a near-optimal, near-feasible level of performance, however, we found that this level of performance is achievable by simple nodewise averaging (Figure 1, Tables 3, 4).

In this work, we introduce **OPFormer-V**, a transformer-based model that treats an N-bus grid as a sequence of N tokens, each encoding node-level features. The encoder outputs are concatenated and passed through a feedforward head to predict all bus voltages. Unlike DeepOPF-V, OPFormer-V can incorporate node-specific but sample-invariant features such as generator limits $(p_{g,i}^{\max}, p_{g,i}^{\min}, q_{g,i}^{\max}, q_{g,i}^{\min})$, generator costs (c_1, c_2), and shunt admittances (bs_i, gs_i).

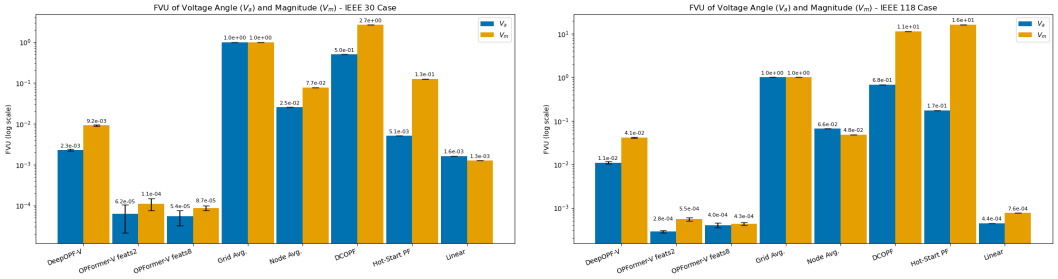

Figure 1: A figure showing the FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the 7 different methods considered for the test split on the IEEE case 30 and case 118 grids. There are 2 variations of the OPFormer-V shown, **feats-2** and **feats-8**

6 Results & Discussion

We evaluated a range of baselines, DeepOPF-V, and our proposed OPFormer-V on multiple datasets using both regression and power metrics. Regression metrics capture accuracy in predicting bus voltages, while power metrics assess whether the resulting generator outputs align with AC-OPF and satisfy constraints. These two views are complementary: improved regression performance does not necessarily translate to an improved feasibility rate or optimal power flow. Figure 1 shows the FVU for voltage magnitude and angle while tables 1 and 2 present the power metrics of DeepOPF-V and OPFormer-V.

Table 1: A table comparing the OPF solutions from DeepOPF-V and OPFormer-V (feats-8) on the test split on the IEEE case30 datasets. Comparing the average relative gap from optimality, the rate of violation of generation limits, the average relative difference between ground truth load and the effective load derived using predicted voltage at both a grid level and at a nodal level for $\neq 0$ loads.

IEEE case30	**DeepOPF-V**		**OPFormer-V**	
Rel. Opt. Diff. (%)	-0.025	$\pm$0.082	**0.087**	$\pm$0.095
Abs. Rel. Opt. Diff. (%)	2.427	$\pm$0.023	**0.150**	$\pm$0.050
$\mathbf{P_g}$ Violation Rate (%)	10.984	$\pm$0.255	**10.488**	$\pm$0.639
$\mathbf{Q_g}$ Violation Rate (%)	**14.932**	$\pm$0.418	16.629	$\pm$1.392
Abs. Rel. Tot. $\mathbf{P_d}$ err. (%)	1.876	$\pm$0.019	**0.116**	$\pm$0.038
Abs. Rel. Tot. $\mathbf{Q_d}$ err. (%)	2.151	$\pm$0.029	**0.144**	$\pm$0.014
Abs. Rel. $\mathbf{P_d^{\neq 0}}$ err. (%)	22.270	$\pm$0.138	**2.251**	$\pm$0.208
Abs. Rel. $\mathbf{Q_d^{\neq 0}}$ err. (%)	23.627	$\pm$0.052	**6.657**	$\pm$0.657

In terms of regression, OPFormer-V achieves the best overall performance, though simple linear baselines such as Nodewise Averaging and Linear Regression are surprisingly competitive, in some cases rivalling or outperforming DeepOPF-V. This suggests that for this data generation process, relatively simple models can capture much of the predictive structure. When we examine the resulting

Table 2: A table comparing the OPF solutions from DeepOPF-V and OPFormer-V (feats-8) on the test split on the IEEE case118 datasets. Comparing the average relative gap from optimality, the rate of violation of generation limits, the average relative difference between ground truth load and the effective load derived using predicted voltage at both a grid level and at a nodal level for $\neq 0$ loads.

IEEE case118	**DeepOPF-V**		**OPFormer-V**	
Rel. Opt. Diff. (%)	-0.618	$\pm$0.012	**-0.153**	$\pm$0.053
Abs. Rel. Opt. Diff. (%)	1.713	$\pm$0.018	**0.323**	$\pm$0.045
$\mathbf{P_g}$ Violation Rate (%)	21.799	$\pm$0.044	**16.468**	$\pm$0.306
$\mathbf{Q_g}$ Violation Rate (%)	12.605	$\pm$0.114	**11.771**	$\pm$0.742
Abs. Rel. Tot. $\mathbf{P_d}$ err. (%)	1.242	$\pm$0.012	**0.245**	$\pm$0.035
Abs. Rel. Tot. $\mathbf{Q_d}$ err. (%)	1.472	$\pm$0.006	**0.241**	$\pm$0.011
Abs. Rel. $\mathbf{P_d^{\neq 0}}$ err. (%)	16.242	$\pm$0.140	**4.053**	$\pm$0.182
Abs. Rel. $\mathbf{Q_d^{\neq 0}}$ err. (%)	17.648	$\pm$0.252	**4.934**	$\pm$0.112

power metrics of DeepOPF-V and OPFormer-V, we see that OPFormer-V consistently achieves lower optimality gaps and reduced errors in effective load. The errors in the effective load reflect the load satisfaction (0% error equals 100% satisfaction). We observed that the aggregate relative error was lower than the average nodal relative error, with this difference being more significant for DeepOPF-V. However, the improvements in voltage prediction did not translate into a proportional decrease in the generator limit violation rate. Our observed generation limit violation rates are higher than those reported in Huang et al. [2021], however, we believe this is because we consider a larger load variation. Overall, OPFormer-V demonstrates strong performance across both regression and power metrics, validating attention-based models as promising alternatives to FCNNs for AC-OPF. Generation constraint violation could be further reduced with a post-processing procedure such as that employed by Huang et al. [2021], however, this step is model agnostic and could be applied to our baseline models as well. The competitive performance of linear baselines shows that the benefits of the direct ML approaches considered are relatively incremental. Full regression and power metric tables and additional model information are provided in the appendix.

References

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

angryavian (https://math.stackexchange.com/users/43949/angryavian). Under what conditions does $e[f(x)] \approx f(e[x])$? Mathematics Stack Exchange, 2019. URL `https://math.stackexchange.com/q/3127971`. URL:https://math.stackexchange.com/q/3127971 (version: 2019-02-26).

Kyri Baker. Emulating ac opf solvers for obtaining sub-second feasible, near-optimal solutions, 2022. URL `https://arxiv.org/abs/2012.10031`.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Balthazar Donon, Rémy Clément, Benjamin Donnot, Antoine Marot, Isabelle Guyon, and Marc Schoenauer. Neural networks for power flow: Graph neural solver. *Electric Power Systems Research*, 189:106547, 2020.

Priya L Donti and J Zico Kolter. Machine learning for sustainable energy systems. *Annual Review of Environment and Resources*, 46:719–747, 2021.

Priya L Donti, Melrose Roderick, Mahyar Fazlyab, and J Zico Kolter. Enforcing robust control guarantees within neural network policies. *arXiv preprint arXiv:2011.08105*, 2020.

Priya L. Donti, David Rolnick, and J. Zico Kolter. Dc3: A learning method for optimization with hard constraints, 2021.

Thomas Falconer and Letif Mones. Deep learning architectures for inference of ac-opf solutions. *arXiv preprint arXiv:2011.03352*, 2020.

Thomas Falconer and Letif Mones. Leveraging power grid topology in machine learning assisted optimal power flow. *IEEE Transactions on Power Systems*, 2022.

Jonas Berg Hansen, Stian Normann Anfinsen, and Filippo Maria Bianchi. Power flow balancing with decentralized graph neural networks. *IEEE Transactions on Power Systems*, 38(3):2423–2433, 2023. doi: 10.1109/TPWRS.2022.3195301.

Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H Low. Deepopf-v: Solving ac-opf problems efficiently. *IEEE Transactions on Power Systems*, 37(1):800–803, 2021.

Meiyi Li, Yuhan Du, Javad Mohammadi, Constance Crozier, Kyri Baker, and Soummya Kar. Numerical comparisons of linear power flow approximations: Optimality, feasibility, and computation time. In *2022 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5, 2022. doi: 10.1109/PESGM48719.2022.9916903.

Zhigang Li, Jinyu Yu, and Q. H. Wu. Approximate linear power flow using logarithmic transform of voltage magnitudes with reactive power and transmission loss consideration. *IEEE Transactions on Power Systems*, 33(4):4593–4603, 2018. doi: 10.1109/TPWRS.2017.2776253.

Heng Liang and Changhong Zhao. Deepopf-u: A unified deep neural network to solve ac optimal power flow in multiple networks, 2023. URL `https://arxiv.org/abs/2309.12849`.

Shaohui Liu, Chengyang Wu, and Hao Zhu. Topology-aware graph neural networks for learning feasible and adaptive ac-opf solutions. *IEEE Transactions on Power Systems*, 2022.

Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Optimal power flow using graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5930–5934, 2020. doi: 10.1109/ICASSP40776.2020.9053140.

Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Unsupervised optimal power flow using graph neural networks. *arXiv preprint arXiv:2210.09277*, 2022.

Thuan Pham and Xingpeng Li. Reduced optimal power flow using graph neural network. In *2022 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2022.

Luis Piloto, Sofia Liguori, Sephora Madjiheurem, Miha Zgubic, Sean Lovett, Hamish Tomlinson, Sophie Elster, Chris Apps, and Sims Witherspoon. Canos: A fast and scalable neural ac-opf solver robust to n-1 perturbations. *arXiv preprint arXiv:2403.17660*, 2024.

Alex Robson, Mahdi Jamei, Cozmin Ududec, and Letif Mones. Learning an optimally reduced formulation of opf through meta-optimization. *arXiv preprint arXiv:1911.06784*, 2019.

Hui Zhang, Gerald T. Heydt, Vijay Vittal, and Jaime Quintero. An improved network model for transmission expansion planning considering reactive power and network losses. *IEEE Transactions on Power Systems*, 28(3):3471–3479, 2013. doi: 10.1109/TPWRS.2013.2250318.

Min Zhou, Minghua Chen, and Steven H. Low. Deepopf-ft: One deep neural network for multiple ac-opf problems with flexible topology. *IEEE Transactions on Power Systems*, 38(1):964–967, 2023. doi: 10.1109/TPWRS.2022.3217407.

Ray D Zimmerman, Carlos E Murillo-Sánchez, and Deqiang Gan. Matpower. *PSERC.[Online]. Software Available at: http://www. pserc. cornell. edu/matpower*, 1997.

A Appendix

A.1 Hot-start linear power flow

Linearised power flow equations using Taylor series and reference point $\tilde{v}$, $\tilde{\delta}$.

$$\begin{aligned} p_{ij} = \tilde{p}_{ij} + 2g_{ij}\tilde{v}_i\Delta_{v_i} + \sin\left(\tilde{\delta}_{ij}\right)\left[g_{ij}\tilde{v}_i\tilde{v}_j\Delta_{\delta_{ij}} - b_{ij}T_{v_i,v_j}\right] \\ - \cos\left(\tilde{\delta}_{ij}\right)\left[g_{ij}T_{v_i,v_j} + b_{ij}\tilde{v}_i\tilde{v}_j\Delta_{\delta_{ij}}\right], \end{aligned} \tag{2}$$

$$\begin{aligned} q_{ij} = \tilde{q}_{ij} - 2b_{ij}\tilde{v}_i\Delta_{v_i} - \sin\left(\tilde{\delta}_{ij}\right)\left[b_{ij}\tilde{v}_i\tilde{v}_j\Delta_{\delta_{ij}} + g_{ij}T_{v_i,v_j}\right] \\ + \cos\left(\tilde{\delta}_{ij}\right)\left[b_{ij}T_{v_i,v_j} - g_{ij}\tilde{v}_i\tilde{v}_j\Delta_{\delta_{ij}}\right], \end{aligned} \tag{3}$$

$$T_{v_i,v_j} = \tilde{v}_i\Delta_{v_j} + \tilde{v}_j\Delta_{v_i}, \tag{4}$$

$$\Delta_{x_k} = x_k - \tilde{x}_k. \tag{5}$$

Upper bound on MAE for hot-start linear power flow.

$$\begin{aligned} R\left(\zeta\right) = g_{ij}\Delta_{v_i}^2 + |y_{ij}|\left[\frac{\zeta_{v_i}\zeta_{v_j}}{2}\cos\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\Delta_{\delta_{ij}}^2 - \Lambda\sin\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\right] \\ \Lambda = \zeta_{v_i}\Delta_{v_j}\Delta_{\delta_{ij}} + \Delta_{v_i}\zeta_{v_j}\Delta_{\delta_{ij}}, \end{aligned} \tag{6}$$

$$\begin{aligned} \mathbb{E}\left[|R\left(\zeta\right)|\right] \leq |g_{ij}|\,\mathbb{E}\left[\Delta_{v_i}^2\right] + |y_{ij}|\left[\frac{v_{ub}^2}{2}\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right] + \mathbb{E}\left[|\Lambda|\right]_{ub}\right] \\ \mathbb{E}\left[|\Lambda|\right]_{ub} = v_{ub}\left[\left(\mathbb{E}\left[\Delta_{v_j}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]\right)^{\frac{1}{2}} + \left(\mathbb{E}\left[\Delta_{v_i}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]\right)^{\frac{1}{2}}\right]. \end{aligned} \tag{7}$$

For a first-order Taylor series approximation of a function the error in approximation is given by the remainder term $R\left(\zeta\right)$ shown in equation 8 where ζ is a point that lies on the line between $\mathbf{x}$ and reference point $\mathbf{a}$ and $\mathbf{H}_\zeta$ is the Hessian evaluated at point ζ.

$$R\left(\zeta\right) = \frac{1}{2}\left(\mathbf{x} - \mathbf{a}\right)^T\mathbf{H}_\zeta\left(\mathbf{x} - \mathbf{a}\right) \tag{8}$$

For active power flow from node i to node j this remainder term takes the form shown in equation 9.

$$\begin{aligned} R\left(\zeta\right) = g_{ij}\Delta_{v_i}^2 + |y_{ij}|\left[\frac{\zeta_{v_i}\zeta_{v_j}}{2}\cos\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\Delta_{\delta_{ij}}^2 - \Lambda\sin\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\right] \\ \Lambda = \zeta_{v_i}\Delta_{v_j}\Delta_{\delta_{ij}} + \Delta_{v_i}\zeta_{v_j}\Delta_{\delta_{ij}} \end{aligned} \tag{9}$$

An upper bound on the absolute value of the remainder can be formed by summing the absolute values of individual terms as seen in equation 10

$$\begin{aligned} |R\left(\zeta\right)| &= \left|g_{ij}\Delta_{v_i}^2 + |y_{ij}|\left[\frac{\zeta_{v_i}\zeta_{v_j}}{2}\cos\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\Delta_{\delta_{ij}}^2 - \Lambda\sin\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\right]\right| \\ &\leq \left|g_{ij}\Delta_{v_i}^2\right| + \left|y_{ij}\frac{\zeta_{v_i}\zeta_{v_j}}{2}\cos\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\Delta_{\delta_{ij}}^2\right| + \left|y_{ij}\Lambda\sin\left(\zeta_{\delta_{ij}} - \angle_{y_{ij}}\right)\right| \\ &\leq \left|g_{ij}\Delta_{v_i}^2\right| + \left|y_{ij}\frac{\zeta_{v_i}\zeta_{v_j}}{2}\Delta_{\delta_{ij}}^2\right| + |y_{ij}\Lambda| \\ &\leq \left|g_{ij}\Delta_{v_i}^2\right| + |y_{ij}|\left[\left|\frac{v_{ub}^2}{2}\Delta_{\delta_{ij}}^2\right| + \left|v_{ub}\Delta_{v_j}\Delta_{\delta_{ij}}\right| + \left|v_{ub}\Delta_{v_i}\Delta_{\delta_{ij}}\right|\right] \end{aligned} \tag{10}$$

If we take the expectation of this upper bound on the remainder we get the first expression in equation 11. If we assume that v_i, v_j and δ_{ij} are independent and symmetric then this expectation is minimised by the mean values of v_i, v_j and δ_{ij}. If we do not want to make this assumption we can use the

Cauchy-Schwartz inequality to find and upper bound on this expectation which is minimised by the mean.

$$\begin{aligned}\mathbb{E}\left[|R\left(\zeta\right)|_{ub}\right] &= |g_{ij}|\,\mathbb{E}\left[\Delta_{v_i}^2\right] + v_{ub}\,|y_{ij}|\left[\frac{v_{ub}\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}{2} + \mathbb{E}\left[\left|\Delta_{v_j}\Delta_{\delta_{ij}}\right|\right] + \mathbb{E}\left[\left|\Delta_{v_i}\Delta_{\delta_{ij}}\right|\right]\right] \\ &\leq |g_{ij}|\,\mathbb{E}\left[\Delta_{v_i}^2\right] + v_{ub}\,|y_{ij}|\left[\frac{v_{ub}\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}{2} + \sqrt{\mathbb{E}\left[\Delta_{v_j}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]} + \sqrt{\mathbb{E}\left[\Delta_{v_i}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}\right]\end{aligned} \tag{11}$$

A similar process can be done for reactive power flow to derive the bound show in equation 12 as b_{ij} is typically much larger g_{ij} we can expect greater error in predicting reactive power flow than active power flow.

$$\begin{aligned}\mathbb{E}\left[|R\left(\zeta\right)|_{ub}\right] &= |b_{ij}|\,\mathbb{E}\left[\Delta_{v_i}^2\right] + v_{ub}\,|y_{ij}|\left[\frac{v_{ub}\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}{2} + \mathbb{E}\left[\left|\Delta_{v_j}\Delta_{\delta_{ij}}\right|\right] + \mathbb{E}\left[\left|\Delta_{v_i}\Delta_{\delta_{ij}}\right|\right]\right] \\ &\leq |b_{ij}|\,\mathbb{E}\left[\Delta_{v_i}^2\right] + v_{ub}\,|y_{ij}|\left[\frac{v_{ub}\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}{2} + \sqrt{\mathbb{E}\left[\Delta_{v_j}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]} + \sqrt{\mathbb{E}\left[\Delta_{v_i}^2\right]\mathbb{E}\left[\Delta_{\delta_{ij}}^2\right]}\right]\end{aligned} \tag{12}$$

A.1.1 Data efficient approximation

Consider the function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^1$ with a Hessian $\mathbf{H}$ that is bounded by $\mathbf{M}$ so that the absolute values of the elements in $\mathbf{H}$ are less than the corresponding element in $\mathbf{M}$, $|\mathbf{H}| \leq \mathbf{M}$. If we examine its Taylor expansion as shown in equation 13 where the reference point is the mean of $\mathbf{x}$ we see that the absolute difference between the value of the function at the mean and the mean of the function value over $\mathbf{x}$ is expressed in terms of the hessian of the function and the covariance of $\mathbf{x}$. These equations extend into the multivariate case the work shown in angryavian [https://math.stackexchange.com/users/43949/angryavian] (Licensed under CC BY-SA 3.0).

$$\begin{aligned}\mathbf{f}(\mathbf{x}) &= \mathbf{f}(\tilde{\mathbf{x}}) + \mathbf{J}_{\mathbf{x}}\left(\mathbf{x}-\tilde{\mathbf{x}}\right) + \frac{1}{2}\left(\mathbf{x}-\tilde{\mathbf{x}}\right)^T\mathbf{H}_{\zeta}\left(\mathbf{x}-\tilde{\mathbf{x}}\right) \\ \mathbf{f}(\mathbf{x}) - \mathbf{f}(\tilde{\mathbf{x}}) &= \mathbf{J}_{\mathbf{x}}\left(\mathbf{x}-\tilde{\mathbf{x}}\right) + \frac{1}{2}\left(\mathbf{x}-\tilde{\mathbf{x}}\right)^T\mathbf{H}_{\zeta}\left(\mathbf{x}-\tilde{\mathbf{x}}\right) \\ \mathbb{E}\left[\mathbf{f}(\mathbf{x})\right] - \mathbf{f}(\tilde{\mathbf{x}}) &= \mathbf{J}_{\mathbf{x}}\left(\mathbb{E}\left[\mathbf{x}\right]-\tilde{\mathbf{x}}\right) + \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_{\zeta}\mathbb{E}\left[\left(\mathbf{x}-\tilde{\mathbf{x}}\right)\left(\mathbf{x}-\tilde{\mathbf{x}}\right)^T\right]\right) \\ &\quad + \frac{1}{2}\left(\mathbb{E}\left[\mathbf{x}\right]-\tilde{\mathbf{x}}\right)^T\mathbf{H}_{\zeta}\left(\mathbb{E}\left[\mathbf{x}\right]-\tilde{\mathbf{x}}\right) \\ \mathbb{E}\left[\mathbf{f}(\mathbf{x})\right] - \mathbf{f}\left(\mathbb{E}\left[\mathbf{x}\right]\right) &= \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_{\zeta}\Sigma_{\mathbf{x}}\right) \\ \left|\mathbb{E}\left[\mathbf{f}(\mathbf{x})\right] - \mathbf{f}\left(\mathbb{E}\left[\mathbf{x}\right]\right)\right| &= \left|\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}_{\zeta}\Sigma_{\mathbf{x}}\right)\right| \\ &\leq \frac{1}{2}\mathrm{Tr}\left(\left|\mathbf{H}_{\zeta}\right|\left|\Sigma_{\mathbf{x}}\right|\right) \\ &\leq \frac{1}{2}\mathrm{Tr}\left(\mathbf{M}\left|\Sigma_{\mathbf{x}}\right|\right)\end{aligned} \tag{13}$$

Numerically, for the self-generated 118 node case we observed, the mean absolute difference over all nodes was $\mathbf{2.1217e-4}$, $\mathbf{4.0167e-4}$ for voltage magnitude and angle, respectively. Numerically, for the self-generated 30 node case we observed, the mean absolute difference over all nodes was $\mathbf{4.5786e-4}$, $\mathbf{9.8582e-4}$ for voltage magnitude and angle, respectively.

A.1.2 Active power generation error comparison

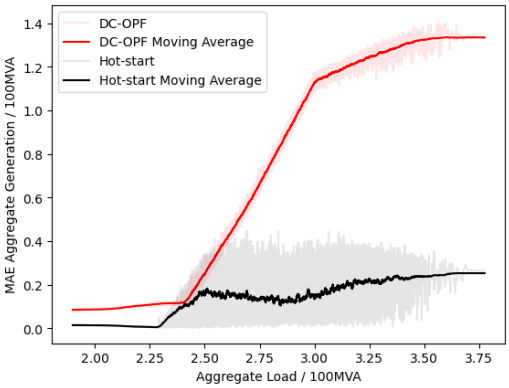

Figure 2: Sum Absolute Error in in active power generation for all generators sorted by aggregate active power demand for the self-generated 30 node case dataset. This figure compares this error in DC-OPF and the hot-start linear power flow. This figure shows error in DC-OPF approximation is typically worse than for hot-start and that this error is dependent on aggregate demand and generally worsens as we increase aggregate demand, saturating at higher levels

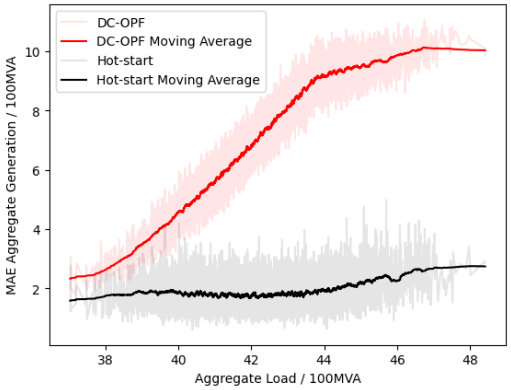

Figure 3: Sum Absolute Error in in active power generation for all generators sorted by aggregate active power demand for the self-generated 118 node case dataset. This figure compares this error in DC-OPF and the hot-start linear power flow. This figure shows error in DC-OPF approximation is typically worse than for hot-start and that this error is dependent on aggregate demand and generally worsens as we increase aggregate demand, saturating at higher levels

A.2 Regression metrics

Table 3: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the 7 different methods considered for the test split on the self-generated dataset on the IEEE case 30 grid. There are 2 variations of the OPFormer-V shown, **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input while **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

Method	V_a		V_m	
	MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	8.782×10^{-6} $\pm 5.3 \times 10^{-7}$	2.280×10^{-3} $\pm 1.4 \times 10^{-4}$	3.272×10^{-6} $\pm 1.2 \times 10^{-7}$	9.169×10^{-3} $\pm 3.3 \times 10^{-4}$
OPFormer-V, feats 2	2.402×10^{-7} $\pm 1.6 \times 10^{-7}$	6.235×10^{-5} $\pm 4.1 \times 10^{-5}$	3.976×10^{-8} $\pm 1.2 \times 10^{-8}$	1.112×10^{-4} $\pm 3.5 \times 10^{-5}$
OPFormer-V, feats 8	$\mathbf{2.089 \times 10^{-7}}$ $\pm 8.4 \times 10^{-8}$	$\mathbf{5.421 \times 10^{-5}}$ $\pm 2.2 \times 10^{-5}$	$\mathbf{3.123 \times 10^{-8}}$ $\pm 4.3 \times 10^{-9}$	$\mathbf{8.731 \times 10^{-5}}$ $\pm 1.2 \times 10^{-5}$
Grid Avg.	3.853×10^{-3}	1.000×10^{-0}	3.577×10^{-4}	1.000×10^{-0}
Node Avg.	9.804×10^{-5}	2.545×10^{-2}	2.765×10^{-5}	7.731×10^{-2}
DC-OPF	1.932×10^{-3}	5.015×10^{-1}	9.493×10^{-4}	2.654×10^{-0}
Hot-Start PF	1.967×10^{-5}	5.105×10^{-3}	4.480×10^{-5}	1.252×10^{-1}
Linear	6.272×10^{-6}	1.628×10^{-3}	4.484×10^{-7}	1.254×10^{-3}

Table 4: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the 7 different methods considered for the test split on the self-generated dataset on the IEEE case 118 grid. There are 2 variations of the OPFormer-V shown, **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input while **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

Method	V_a		V_m	
	MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	7.351×10^{-5}	1.090×10^{-2}	6.458×10^{-6}	4.129×10^{-2}
	$\pm 4.4 \times 10^{-6}$	$\pm 6.5 \times 10^{-4}$	$\pm 9.5 \times 10^{-8}$	$\pm 6.1 \times 10^{-4}$
OPFormer-V, feats 2	$\mathbf{1.921 \times 10^{-6}}$	$\mathbf{2.849 \times 10^{-4}}$	8.545×10^{-8}	5.464×10^{-4}
	$\pm 1.3 \times 10^{-7}$	$\pm 1.9 \times 10^{-5}$	$\pm 8.1 \times 10^{-9}$	$\pm 5.2 \times 10^{-5}$
OPFormer-V, feats 8	2.708×10^{-6}	4.016×10^{-4}	$\mathbf{6.703 \times 10^{-8}}$	$\mathbf{4.286 \times 10^{-4}}$
	$\pm 3.2 \times 10^{-7}$	$\pm 4.7 \times 10^{-5}$	$\pm 5.4 \times 10^{-9}$	$\pm 3.4 \times 10^{-5}$
Grid Avg.	6.743×10^{-3}	1.000×10^{-0}	1.564×10^{-4}	1.000×10^{-0}
Node Avg.	4.447×10^{-4}	6.595×10^{-2}	7.528×10^{-6}	4.814×10^{-2}
DC-OPF	4.575×10^{-3}	6.785×10^{-1}	1.795×10^{-3}	$1.148 \times 10^{+1}$
Hot-Start PF	1.174×10^{-3}	1.741×10^{-1}	2.519×10^{-3}	$1.611 \times 10^{+1}$
Linear	2.986×10^{-6}	4.428×10^{-4}	1.188×10^{-7}	7.596×10^{-4}

Table 5: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the different methods considered for the train split on the self-generated dataset on the IEEE case 30 grid. There are 2 variations of the OPFormer-V shown, **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input while **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

IEEE case30 (Train)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	8.746×10^{-6}	2.275×10^{-3}	3.260×10^{-6}	9.128×10^{-3}
	(σ)	$\pm 5.3 \times 10^{-7}$	$\pm 1.4 \times 10^{-4}$	$\pm 1.3 \times 10^{-7}$	$\pm 3.5 \times 10^{-4}$
OPFormer-V	(μ)	2.077×10^{-7}	5.402×10^{-5}	3.115×10^{-8}	8.723×10^{-5}
(feats 8)	(σ)	$\pm 8.3 \times 10^{-8}$	$\pm 2.2 \times 10^{-5}$	$\pm 4.3 \times 10^{-9}$	$\pm 1.2 \times 10^{-5}$
OPFormer-V	(μ)	2.393×10^{-7}	6.225×10^{-5}	3.964×10^{-8}	1.110×10^{-4}
(feats 2)	(σ)	$\pm 1.6 \times 10^{-7}$	$\pm 4.1 \times 10^{-5}$	$\pm 1.2 \times 10^{-8}$	$\pm 3.5 \times 10^{-5}$
Grid Avg.		3.845×10^{-3}	1.000×10^{-0}	3.571×10^{-4}	1.000×10^{-0}
Node Avg.		9.768×10^{-5}	2.540×10^{-2}	2.726×10^{-5}	7.633×10^{-2}
DC-OPF		1.925×10^{-3}	5.007×10^{-1}	9.490×10^{-4}	2.657×10^{-0}
Linear		6.328×10^{-6}	1.646×10^{-3}	4.464×10^{-7}	1.250×10^{-3}
GP		1.225×10^{-17}	3.186×10^{-15}	4.082×10^{-11}	1.143×10^{-7}
Hot-Start		1.899×10^{-5}	4.939×10^{-3}	4.484×10^{-5}	1.256×10^{-1}

Table 6: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the different methods considered for the validation split on the self-generated dataset on the IEEE case 30 grid. There are 2 variations of the OPFormer-V shown, **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input while **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

IEEE case30 (Val.)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	8.778×10^{-6}	2.282×10^{-3}	3.281×10^{-6}	9.204×10^{-3}
	(σ)	$\pm5.3\times10^{-7}$	$\pm1.4\times10^{-4}$	$\pm1.2\times10^{-7}$	$\pm3.5\times10^{-4}$
OPFormer-V (feats 8)	(μ)	2.139×10^{-7}	5.560×10^{-5}	3.208×10^{-8}	9.001×10^{-5}
	(σ)	$\pm8.5\times10^{-8}$	$\pm2.2\times10^{-5}$	$\pm4.4\times10^{-9}$	$\pm1.2\times10^{-5}$
OPFormer-V (feats 2)	(μ)	2.413×10^{-7}	6.272×10^{-5}	4.036×10^{-8}	1.132×10^{-4}
	(σ)	$\pm1.6\times10^{-7}$	$\pm4.0\times10^{-5}$	$\pm1.3\times10^{-8}$	$\pm3.6\times10^{-5}$
Grid Avg.		3.847×10^{-3}	1.000×10^{-0}	3.564×10^{-4}	1.000×10^{-0}
Node Avg.		9.799×10^{-5}	2.547×10^{-2}	2.735×10^{-5}	7.674×10^{-2}
DC-OPF		1.915×10^{-3}	4.977×10^{-1}	9.504×10^{-4}	2.666×10^{-0}
Linear		6.538×10^{-6}	1.700×10^{-3}	4.604×10^{-7}	1.292×10^{-3}
GP		1.200×10^{-6}	3.118×10^{-4}	1.002×10^{-7}	2.811×10^{-4}
Hot-Start		1.900×10^{-5}	4.939×10^{-3}	4.446×10^{-5}	1.247×10^{-1}

Table 7: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the different methods considered for the train split on the self-generated dataset on the IEEE case 118 grid. The OPFormer-V variation considered **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

IEEE case118 (Train)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	7.374×10^{-5}	1.093×10^{-2}	6.473×10^{-6}	4.140×10^{-2}
	(σ)	$\pm4.3\times10^{-6}$	$\pm6.4\times10^{-4}$	$\pm9.7\times10^{-8}$	$\pm6.2\times10^{-4}$
OPFormer-V (feats 8)	(μ)	2.661×10^{-6}	3.943×10^{-4}	6.570×10^{-8}	4.202×10^{-4}
	(σ)	$\pm3.1\times10^{-7}$	$\pm4.7\times10^{-5}$	$\pm5.4\times10^{-9}$	$\pm3.4\times10^{-5}$
Grid Avg.		6.749×10^{-3}	1.000×10^{-0}	1.563×10^{-4}	1.000×10^{-0}
Node Avg.		4.428×10^{-4}	6.560×10^{-2}	7.550×10^{-6}	4.829×10^{-2}
DC-OPF		4.573×10^{-3}	6.775×10^{-1}	1.795×10^{-3}	$1.148\times10^{+1}$
Linear		2.948×10^{-6}	4.368×10^{-4}	1.178×10^{-7}	7.534×10^{-4}
GP		6.290×10^{-6}	9.320×10^{-4}	2.595×10^{-7}	1.660×10^{-3}
Hot-Start		1.172×10^{-3}	1.737×10^{-1}	2.526×10^{-3}	$1.616\times10^{+1}$

A.3 Power metrics

For the linear power flow methods full AC-OPF solutions were also generated by using predicted voltages in the power flow equations, hence the violation in generation. An alternative approach could use predicted generator output and automatically satisfy generation constraints, but result in potential voltage violations.

Table 8: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu of the different methods considered for the validation split on the self-generated dataset on the IEEE case 118 grid. The OPFormer-V variation considered **feats-8** takes an eight dimensional vector of load, shunt susceptance and generator information $(p_{l,i}, q_{l,i}, bs_i, p_{g,i}^{max}, q_{g,i}^{max}, q_{g,i}^{min}, c1, c2)$. For NN methods we report the mean and standard deviation over 3 runs.

IEEE case118 (Val.)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	7.326×10^{-5}	1.085×10^{-2}	6.465×10^{-6}	4.133×10^{-2}
	(σ)	$\pm4.1\times10^{-6}$	$\pm6.1\times10^{-4}$	$\pm9.4\times10^{-8}$	$\pm6.0\times10^{-4}$
OPFormer-V	(μ)	2.774×10^{-6}	4.110×10^{-4}	6.755×10^{-8}	4.319×10^{-4}
(feats 8)	(σ)	$\pm3.1\times10^{-7}$	$\pm4.5\times10^{-5}$	$\pm5.7\times10^{-9}$	$\pm3.6\times10^{-5}$
Grid Avg.		6.750×10^{-3}	1.000×10^{-0}	1.564×10^{-4}	1.000×10^{-0}
Node Avg.		4.445×10^{-4}	6.585×10^{-2}	7.541×10^{-6}	4.821×10^{-2}
DC-OPF		4.540×10^{-3}	6.726×10^{-1}	1.795×10^{-3}	$1.147\times10^{+1}$
Linear		3.049×10^{-6}	4.516×10^{-4}	1.191×10^{-7}	7.612×10^{-4}
GP		8.556×10^{-6}	1.267×10^{-3}	3.804×10^{-7}	2.432×10^{-3}
Hot-Start		1.196×10^{-3}	1.772×10^{-1}	2.526×10^{-3}	$1.615\times10^{+1}$

Table 9: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu for the different methods considered on the train split on the OPF-Learn case 30 dataset. The OPFormer-V variation considered **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input. For NN methods we report the mean and standard deviation over 3 runs.

OPF-Learn case30 (Train)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	4.227×10^{-7}	1.368×10^{-1}	2.384×10^{-4}	2.762×10^{-1}
	(σ)	$\pm2.3\times10^{-12}$	$\pm7.4\times10^{-7}$	$\pm2.0\times10^{-9}$	$\pm2.3\times10^{-6}$
OPFormer-V	(μ)	3.388×10^{-7}	1.096×10^{-1}	2.194×10^{-4}	2.541×10^{-1}
(feats 2)	(σ)	$\pm7.2\times10^{-8}$	$\pm2.3\times10^{-2}$	$\pm1.7\times10^{-5}$	$\pm2.0\times10^{-2}$
Grid Avg.		3.091×10^{-6}	1.000×10^{-0}	8.632×10^{-4}	1.000×10^{-0}
Node Avg.		4.227×10^{-7}	1.368×10^{-1}	2.384×10^{-4}	2.762×10^{-1}
Linear		6.426×10^{-9}	2.079×10^{-3}	3.350×10^{-5}	3.881×10^{-2}
GP		5.099×10^{-9}	1.650×10^{-3}	2.170×10^{-5}	2.514×10^{-2}

Table 10: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu for the different methods considered on the validation split on the OPF-Learn case 30 dataset. The OPFormer-V variation considered **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input. For NN methods we report the mean and standard deviation over 3 runs.

OPF-Learn case30 (Val.)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	4.278×10^{-7}	1.374×10^{-1}	2.468×10^{-4}	2.831×10^{-1}
	(σ)	$\pm4.0\times10^{-12}$	$\pm1.3\times10^{-6}$	$\pm4.3\times10^{-9}$	$\pm4.9\times10^{-6}$
OPFormer-V	(μ)	3.412×10^{-7}	1.096×10^{-1}	2.268×10^{-4}	2.602×10^{-1}
(feats 2)	(σ)	$\pm7.4\times10^{-8}$	$\pm2.4\times10^{-2}$	$\pm1.8\times10^{-5}$	$\pm2.1\times10^{-2}$
Grid Avg.		3.115×10^{-6}	1.000×10^{-0}	8.715×10^{-4}	1.000×10^{-0}
Node Avg.		4.278×10^{-7}	1.374×10^{-1}	2.468×10^{-4}	2.831×10^{-1}
Linear		6.617×10^{-9}	2.124×10^{-3}	3.336×10^{-5}	3.828×10^{-2}
GP		6.680×10^{-9}	2.145×10^{-3}	3.431×10^{-5}	3.937×10^{-2}

Table 11: A table showing the MSE and FVU in predicting the voltage angle (V_a) in rad and voltage magnitude (V_m) in pu for the different methods considered on the test split on the OPF-Learn case 30 dataset. The OPFormer-V variation considered **feats-2** takes a 2 dimensional vector of load $(p_{l,i}, q_{l,i})$ as input. For NN methods we report the mean and standard deviation over 3 runs.

OPF-Learn case30 (Test)		V_a		V_m	
		MSE	**FVU**	**MSE**	**FVU**
DeepOPF-V	(μ)	4.334×10^{-7}	1.393×10^{-1}	2.396×10^{-4}	2.750×10^{-1}
	(σ)	$\pm 5.8 \times 10^{-12}$	$\pm 1.9 \times 10^{-6}$	$\pm 1.3 \times 10^{-9}$	$\pm 1.5 \times 10^{-6}$
OPFormer-V	(μ)	3.458×10^{-7}	1.112×10^{-1}	2.205×10^{-4}	2.530×10^{-1}
(feats 2)	(σ)	$\pm 7.6 \times 10^{-8}$	$\pm 2.4 \times 10^{-2}$	$\pm 1.7 \times 10^{-5}$	$\pm 2.0 \times 10^{-2}$
Grid Avg.		3.111×10^{-6}	1.000×10^{0}	8.715×10^{-4}	1.000×10^{0}
Node Avg.		4.334×10^{-7}	1.393×10^{-1}	2.396×10^{-4}	2.749×10^{-1}
Linear		$\mathbf{7.404 \times 10^{-9}}$	$\mathbf{2.380 \times 10^{-3}}$	$\mathbf{3.475 \times 10^{-5}}$	$\mathbf{3.987 \times 10^{-2}}$
GP		7.555×10^{-9}	2.428×10^{-3}	3.532×10^{-5}	4.053×10^{-2}

Table 12: A table comparing the quality of the OPF solutions from the predictions of the other methods considered on the test split on the IEEE case30 datasets. Predictions are assessed on the relative gap from optimality, the rate of violation of generation limits, the relative difference between load in the ground truth and effective load derived using predicted voltage for both a grid aggregation and at a nodal level for $\neq 0$ loads.

IEEE case30	**Grid**	**Node**	**DC-OPF**	**OLS**	**GP**	**Hot-Start**
Rel. Opt. Diff. (%)	37.373	-0.327	6.383	0.005	-0.002	-0.033
Abs. Rel. Opt. Diff. (%)	37.373	4.271	6.383	0.029	0.087	0.143
$\mathbf{P_g}$ Violation Rate (%)	23.967	4.467	2.097	9.097	11.323	7.796
$\mathbf{Q_g}$ Violation Rate (%)	17.949	24.149	32.979	15.222	13.606	28.670
Abs. Rel. Tot. $\mathbf{P_d}$ (%)	48.987	3.284	5.931	0.018	0.066	0.122
Abs. Rel. Tot. $\mathbf{Q_d}$ (%)	22.308	3.892	73.524	0.088	0.067	0.912
Abs. Rel. $\mathbf{P_d^{\neq 0}}$ (%)	85.714	24.660	29.461	0.197	0.242	0.258
Abs. Rel. $\mathbf{Q_d^{\neq 0}}$ (%)	248.173	24.760	431.436	0.326	0.309	1.733

Table 13: A table comparing the quality of the OPF solutions from the predictions of the other methods considered on the test split on the IEEE case118 datasets. Predictions are assessed on the relative gap from optimality, the rate of violation of generation limits, the relative difference between load in the ground truth and effective load derived using predicted voltage for both a grid aggregation and at a nodal level for $\neq 0$ loads.

IEEE case30	**Grid**	**Node**	**DC-OPF**	**OLS**	**GP**	**Hot-Start**
Rel. Opt. Diff. (%)	14.904	-0.831	2.057	0.002	-0.107	-0.378
Abs. Rel. Opt. Diff. (%)	14.904	1.772	2.057	0.012	2.212	0.382
$\mathbf{P_g}$ Violation Rate (%)	16.953	21.645	4.221	15.201	16.509	14.025
$\mathbf{Q_g}$ Violation Rate (%)	14.437	13.263	27.334	10.691	11.914	28.673
Abs. Rel. Tot. $\mathbf{P_d}$ (%)	33.798	1.254	2.186	0.009	1.673	0.381
Abs. Rel. Tot. $\mathbf{Q_d}$ (%)	26.182	1.472	60.329	0.088	1.715	1.548
Abs. Rel. $\mathbf{P_d^{\neq 0}}$ (%)	54.545	15.707	6.520	0.100	3.171	1.383
Abs. Rel. $\mathbf{Q_d^{\neq 0}}$ (%)	90.601	17.028	149.715	0.467	5.175	4.328

A.4 Additional experiment Information

A.4.1 Transformer

:

- **input size**: 8 or 2
- **num. layers**: 7
- **num. transformer encoder layers**: 4
- **dim. ff**: 512
- **num. attn. heads**: 4
- **c hidden**: 16
- **c out**: 2 * (num. nodes)
- **dropout rate**: 0.1
- **num. parameters:** $\sim$104k (30 nodes), 574k (118 nodes)

A.4.2 MLP

:

- **input size**: 2 * (num. nonzero load nodes)
- **num. layers**: 8
- **c hidden**: 256 (for 30 nodes), 1024 (for 118 nodes)
- **c out**: 2 * (num. nodes)
- **dropout rate**: 0.1
- **num. parameters:** $\sim$359k (30 nodes), 5.7M (118 nodes)

A.4.3 Optimizer details

- **optimizer**: SGD
- **learning rate**: 1e-3
- **weight decay**: 2e-6
- **momentum**: 0.9
- **lr scheduler**: Cosine Annealing
- **num. epochs**: 200

A.4.4 Compute resources & approximate run times

Models trained on CPU (Apple M1 Pro Chip). For the 30 node case approximate train time of 2 hours and the 118 node case approximate train time of 5 hours. OPFLearn case30 approximate train time of 0.5 hours.

A.4.5 Speed-ups, MAC & approximate parameter count

In Tables 14 and 15, we report speed-ups, multiply–accumulate operations (MAC), and parameter counts observed in our experiments. Note that the AC-OPF solver was run on online resources (MATLAB Online), while ML models were trained on a CPU (Apple M1 Pro, 16GB RAM).

Metric	**DeepOPF-V**	**OPFormer-V (feats 2)**
MAC	5.351M	441K
Parameter Count	5.357M	43.2K
Approx. Speedup	$\times$446	$\times$717

Table 14: Estimated MAC, parameter count, and speedup of DeepOPF-V and OPFormer-V (feats 2) on the 30-node case.

Metric	**DeepOPF-V**	**OPFormer-V (feats 2)**
MAC	5.743M	2.891M
Parameter Count	5.750M	449K
Approx. Speedup	$\times$553	$\times$257

Table 15: Estimated MAC, parameter count, and speedup of DeepOPF-V and OPFormer-V (feats 2) on the 118-node case.