Generating adversarial inputs for a graph neural network model of AC power flow

Robert Parker

Los Alamos National Laboratory

Suggested Citation

R. Parker, "Generating adversarial inputs for a graph neural network model of AC power flow," in *Proceedings of the PowerUp Conference*, 2026.

BibTeX Entry

```
@inproceedings{parker2026generatingadversaria,
  author = {Parker, Robert},
  title = {Generating adversarial inputs for a graph neural network model of {AC}
      power flow},
  booktitle = {Proceedings of the PowerUp Conference},
  year = {2026},
}
```

Generating adversarial inputs for a graph neural network model of AC power flow

Robert Parker†

***Abstract*—This work formulates and solves optimization problems to generate input points that yield high errors between a neural network's predicted AC power flow solution and solutions to the AC power flow equations. We demonstrate this capability on an instance of the CANOS-PF graph neural network model, as implemented by the PFΔ benchmark library, operating on a 14-bus test grid. Generated adversarial points yield errors as large as 3.7 per-unit in reactive power and 0.08 per-unit in voltage magnitude. When minimizing the perturbation from a training point necessary to satisfy adversarial constraints, we find that the constraints can be met with as little as an 0.04 per-unit perturbation in voltage magnitude on a single bus. This work motivates the development of rigorous verification and robust training methods for neural network surrogate models of AC power flow.**

***Index Terms*—AC Power Flow, Neural Networks, Optimization**

I. Introduction

Recent work has focused on developing neural network surrogate models to approximate solutions to the alternating current power flow (ACPF) equations. Different architectures such as physics-informed neural networks (PINNs) [1], [2] and graph neural networks (GNNs) [3]–[5] have been proposed and datasets such as OPF-Learn [6], OPFData [7], and PFΔ [8] have been developed to standardize comparisons among different neural (optimal) power flow solvers. These neural network models achieve fast online inference in exchange for a (usually) small approximation error and large offline training cost. Fast online inference is advantageous in the power transmission setting where dispatch decisions are made frequently and network topologies are generally static. However, it is well-known that neural networks are not robust to adversarial perturbations [9], [10]. In safety-critical fields such as electric power transmission, robustness of neural network-based algorithms is especially important. This work addresses the following question: Given a particular surrogate model for AC power flow, can we find input points that have large errors between the surrogate's output and a "ground truth" solution of the AC power flow equations?

Our contribution is to formulate and solve two types of optimization problems for generating adversarial input points for an instance of the CANOS-PF graph neural network ACPF surrogate operating on the IEEE 14-bus test system from PGLib-OPF [11] and trained on data from PFΔ. (See [5] for details on the CANOS architecture; we use the open-source CANOS-PF GNN model implemented by PFΔ [8].) We demonstrate that these optimization problems are tractable for a state-of-the-art embedded neural network model and that, for this model, there exist adversarial points that introduce significant discrepancies between the NN and ACPF solutions. These points introduce large deviations in output variables that are critical for grid operators to predict accurately, such as reactive powers and voltage magnitudes. These results motivate further research into verification and robust training methods for neural network surrogates of AC power flow.

†Los Alamos National Laboratory, Los Alamos, NM, USA
LA-UR-26-20748

A. Related work

Many works have addressed the issue of surrogate model robustness in power system operation. Chen et al. [12] have identified adversarial inputs that result in misclassifications and forecasting errors, while Dinh et al. [13] have analyzed the input points that cause a multi-layer perceptron (MLP) model to make inaccurate predictions. Chevalier et al. have identified loading conditions that cause (1) a DCOPF solution to be AC-infeasible [14] and (2) an MLP's line switching decisions to cause high load shedding [15]. Simultaneously, methods for verification [2], robust training [16], and guaranteeing feasibility at inference time [17] of neural networks for AC power flow have been developed. In contrast to the "adversarial input" papers above, our work targets a graph neural network model that is state-of-the-art (according to a recent benchmark [8]). We also impose constraints on the neural network's outputs (see Problem 2), which has only been done by [15].

II. Problem formulation

We formulate and solve two types of optimization problems: *Maximum-error* problems that maximize the difference between specified coordinates of the neural network's output and ACPF solution and *constrained-error* problems that find the smallest input perturbations that satisfy constraints on each output. Each problem instance targets a specific output, such as voltage magnitude on a PV bus. In constrained-error problems, we constrain a particular output to be "sufficiently different" between the NN and ACPF solutions.

The maximum-error problem is given by Problem 1:

$$\max \ \ (y_{\mathrm{NN},i} - y_{\mathrm{PF},i}) \ \text{ such that } \begin{cases} y_{\mathrm{NN}} = \mathrm{NN}(x) \\ y_{\mathrm{PF}} = \mathrm{PF}(x) \\ L \leq x \leq U \end{cases} \tag{1}$$

Here, x is a vector of inputs to the AC power flow equations: Active power injections and voltage magnitudes on PV buses, active and reactive powers on PQ buses, and voltage angle and magnitude on the reference bus. Function NN evaluates the neural network. Vector y_{NN} contains the outputs predicted by the neural network: Reactive powers and voltage angles at PV buses, voltage angles and magnitudes at PQ buses, and active and reactive power at the reference bus. Function PF solves the AC power flow equations and returns y_{PF}, the same

outputs, now determined by these equations. Bounds on the input vector ensure this problem remains bounded. Problem 1 maximizes the difference between the i-th coordinate of the neural network's output and the i-th coordinate of the AC power flow equations' output. We solve instances of Problem 1 with both maximization and minimization objective senses to maximize both the positive and negative errors between the neural network's output and the power flow equations' output.

The constrained-error problem is given by Problem 2:

$$\min \ \|x - x_0\|_1 \ \text{ such that } \begin{cases} y_{\text{NN}} = \text{NN}(x); & y_{\text{NN},i} \geq l_i \\ y_{\text{PF}} = \text{PF}(x); & y_{\text{PF},i} \leq l_i - \delta \\ L \leq x \leq U \end{cases} \tag{2}$$

Vector x_0 is a reference point obtained from the training data of our surrogate model. We use constraints to enforce a minimum difference specified by the constant parameter δ between the i-th coordinate of the neural network and power flow output vectors. Constant parameter l_i emulates a lower bound that would typically be applied to this output in an optimal power flow problem, such as a lower bound on a voltage magnitude output. Adversarial points identified by Problem 2 can be interpreted as points where the neural network predicts a feasible output (according to the bound l_i in the selected coordinate) but solving the AC power flow equations yields a solution that violates this bound by a margin of at least δ. While we define this problem as targeting a single lower bound l_i, the problem could easily be adjusted to target any number of lower or upper bounds.

III. Test Problem

We solve instances of Problems 1 and 2 with an implementation of the CANOS-PF graph neural network operating on a 14-bus test grid. We use the hyperparameters suggested by [8]: A hidden dimension of 128, 15 message passing steps, a learning rate of 10^{-5}, and 50 training epochs. The model has approximately 7.5 million trainable parameters.

We use the IEEE 14-bus test network from PGLib-OPF [11]. Input bounds are taken from the default PGLib test case. In the constrained-error problem, our output constraints target voltage magnitude at a specified PQ bus. We use the lower bound provided by the PGLib test case and a margin of $\delta = 0.04$ (per-unit). While the input vector, x, contains net loads on PQ buses, we use the convention that loads are non-dispatchable and are therefore fixed to their nominal values (the default PGLib values in the case of Problem 1 and the values from the training data point x_0 in the case of Problem 2). Similarly, we fix inputs corresponding to reference bus voltage angle and magnitude to zero and 1.0 (per-unit). This significantly constrains the feasible space of these optimization problems. Degrees of freedom are net active power injections p^{net} and voltage magnitudes v at PV buses.

IV. Results

We use PFΔ's CANOS-PF model implementation, dataset, and training script. We use the Task-1.1 training dataset, which provides 48,600 input and output points representing solved power flow instances on the full network topology (no contingencies). We implement the optimization problem using JuMP [18] and MathProgIncidence.jl [19]. PowerModels [20] implements the AC power flow constraints and MathOptAI.jl [21] implements the neural network constraints by interfacing with PyTorch [22] via a GPU-accelerated gray-box formulation [23]. We solve optimization problems (locally) with IPOPT [24] using MA57 [25] as the linear solver, a tolerance of 10^{-6}, an "acceptable tolerance" of 10^{-4}, and an iteration limit of 500. All computational experiments, including neural network training, were performed on the Selene supercomputer at Los Alamos National Laboratory using an Intel 8470 CPU and NVIDIA H100 GPU. The code used to produce these results is available at https://github.com/Robbybp/pfdelta.

TABLE I
Losses obtained by our trained CANOS-PF model

Dataset	N. points	MSE			PBL		
		Mean	Stdev.	Max.	Mean	Stdev.	Max.
Train	48,600	3e-4	2e-5	4e-4	1.6e-2	2e-4	1.7e-2
Adversarial	87	0.7	1.0	5.7	0.3	0.1	0.6

A. Neural network training results

We train the CANOS-PF model with the PFΔ Task-1.1 training dataset and script for 50 epochs with a learning rate of 5×10^{-4}. Training uses a combination of mean-squared-error (MSE) loss, a supervised measure, and "constraint violation loss," an unsupervised measure that penalizes violation of the AC power flow equations. See [26] for details behind this combined loss and [5] for the particular implementation in CANOS. Following [8], we evaluate the quality of our trained neueral network with MSE loss and "power balance loss" (PBL), an unsupervised measure that penalizes only violation of active and reactive power balance constraints.

Loss values on train and test data are shown in Table I. These values are comparable to MSE and PBL losses reported by [8], so we conclude that we have accurately reproduced this work's CANOS-PF model. (Compare our training MSE loss to Table A.7 and our PBL to Table A.8 in [8]. While neither is a perfect comparison, our trained model's loss is lower than those presented in [8] in both categories, as expected for *training* loss and our smaller network.) The "Adversarial" dataset in Table I refers to the adversarial points generated by solving Problems 1 and 2: 23 points from Problem 1 and 64 points from Problem 2. The generation of these points is discussed in Sections IV-B and IV-C.

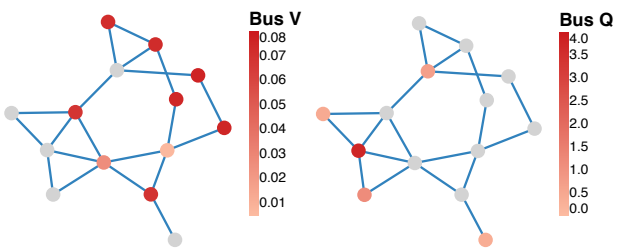

Fig. 1. Maximum absolute error obtained for each bus. Left: PQ buses. Right: PV and reference buses. In each case, the opposite set of buses is shaded gray. Generated using PowerPlots [27].

B. Maximum-error results

We solve two instances of Problem 1 for each bus: One maximizing $(y_{\text{NN},i} - y_{\text{PF},i})$ and one minimizing the same quantity. For PQ buses, the target output (coordinate i) is voltage magnitude, v; for PV and reference buses, the target output is reactive power, q. These errors we achieve are illustrated in Figure 1, where the maximum absolute values between maximization and minimization errors are displayed. The results for both maximization and minimization problems are displayed in Figure 2.

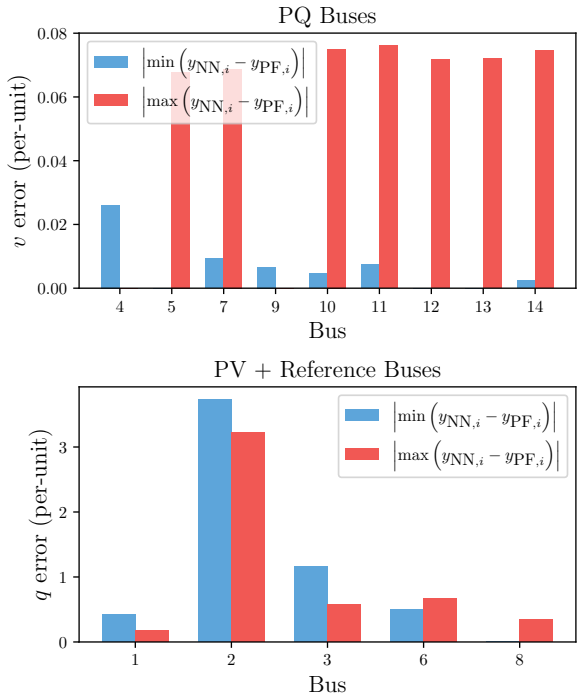

Fig. 2. Errors achieved by Problem 1 with maximization and minimization objectives. Bars are omitted where the solver fails to converge.

The results indicate that large errors between the neural network and AC power flow solutions can be found. Voltage magnitude errors appear to be systematically larger when maximizing than when minimizing, suggesting that the neural network has difficulty approximating solutions accurately when the power flow solution yields a relatively low voltage.

C. Constrained-error results

We solve instances of Problem 2 for each of nine PQ buses and for the first ten training points of the PFΔ Task-1.1 dataset. Table II summarizes the results by training point. We first note that all PQ buses have voltage magnitude lower bounds (according to PGLib) of 0.94, so our constraint on $y_{\text{PF},i}$ (with margin $\delta = 0.04$) is always $y_{\text{PF},i} \leq 0.90$. The errors in voltage magnitude we achieve range from 0.04 to 0.06. There is a large variation in the number of successfully converged problems for different training points, suggesting that some training points are more vulnerable to adversarial perturbations than others. Overall, 64 / 90 instances converge to primal-feasible points.

The "0-norm" in Table II is the number of nonzero entries of its argument, i.e., the number of input variables that need to be perturbed to achieve the adversarial outcome.

TABLE II
SUMMARY OF ADVERSARIAL PERTURBATIONS BY TRAINING POINT

Training point index	Converged (of 9)	Average $\|x - x_0\|_1$	Average $\|x - x_0\|_0$	Average $y_{\text{NN},i}$	Average $y_{\text{PF},i}$
0	9	0.31	3	0.96	0.90
1	4	0.29	3	0.96	0.90
2	8	0.14	3	0.96	0.90
3	6	0.11	2	0.96	0.90
4	6	0.10	2	0.96	0.90
5	9	0.21	2	0.95	0.90
6	8	0.13	2	0.95	0.90
7	6	0.26	3	0.95	0.90
8	3	0.09	2	0.94	0.90
9	5	0.18	2	0.96	0.90

TABLE III
ADVERSARIAL PERTURBATIONS FOR SELECTED CASES

Training point	Targeted bus	$\|x - x_0\|_1$	$\|x - x_0\|_0$	Variable values
4	12	0.04	1	$v_6 = 0.97$
0	4	0.18	2	$v_2 = 0.94,\ v_3 = 0.95$
2	5	0.10	3	$v_1 = 0.94,\ v_2 = 0.94,$ $v_6 = 1.03$
7	7	0.41	4	$v_2 = 0.94,\ v_3 = 0.95,$ $v_6 = 0.94,\ v_8 = 0.94$
0	13	1.03	6	$v_1 = 0.94,\ p_2^{\text{net}} = -0.02$ $v_2 = 0.94,\ v_3 = 0.94,$ $v_6 = 0.94,\ v_8 = 0.94$

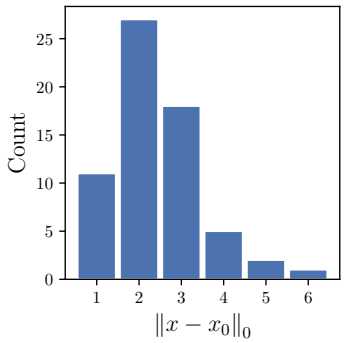

Fig. 3. Histogram of 0-norm of perturbations necessary to satisfy adversarial constraints for the 64 converged instances of Problem 2.

We note that these numbers are low. As shown by the histogram in Figure 3, the adversarial outcome is achieved by perturbing two or fewer input variables in most cases. Selected adversarial perturbations are shown in Table III. We note that each variable listed corresponds to a single perturbed coordinate of x. As shown in Table III, these perturbations often require setting voltage magnitudes on PV buses to values close to their lower bounds of 0.94. This result suggests that robustness of the CANOS-PF neural network model could be improved by including more points with output voltages near this bound in the training data.

D. Computational details

Table IV reports information about the optimization problems solved. Iteration counts and solve times are averages over the converged cases where "converged" is defined as

TABLE IV
STATISTICS FROM OPTIMIZATION PROBLEM SOLVES

Problem	Converged	Var.	Con.	JNNZ	HNNZ	Iter.	Time (s)
(1)	23 / 28	470	481	30k	24k	168	92
(2)	64 / 90	902	699	30k	24k	25	14

terminating at a primal-feasible point within 500 iterations. Computational cost is dominated by function and derivative evaluations, which we assume is due to the cost of evaluating and differentiating the CANOS-PF model at every iteration. These solve times are relatively high despite offloading CANOS-PF evaluation to a GPU. Fast optimization solves with large, dense, and sequential NNs embedded have been demonstrated [23]. However, it is not clear whether these results should extend to the more intricate CANOS GNN architecture. Scalability of Problems 1 and 2 will depend on how efficiently GNNs such as CANOS can be differentiated at larger scales.

Finally, we note that the CANOS model violates the assumptions of the IPOPT solver due to its many non-differentiable ReLU layers. It is surprising that we can converge 74% of the problems attempted despite this apparent incompatibility.

V. Conclusion

This work is a proof-of-concept that adversarial points can be systematically identified for a state-of-the-art neural network model of AC power flow. We have demonstrated the method on a small, 14-bus test system. Future work should apply these methods to larger networks and additional NN models. We note that by keeping loads fixed, we have significantly constrained the input space that we search for adversarial points. Worse points than those shown here can likely be generated if loads (and line parameters) are used as degrees of freedom as well. This work motivates the need for continued robustness testing of neural network surrogate models for AC power flow, development of rigorous validation methods for these models, and development of adversarially robust training and inference methodologies.

Acknowledgements

AI was used to write scripts to generate some of the figures and tables presented in this work after the raw data had been generated by a human programmer.

References

[1] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow," *Electric Power Systems Research*, vol. 212, p. 108412, 2022.

[2] J. Jalving, M. Eydenberg, L. Blakely, A. Castillo, Z. Kilwein, J. K. Skolfield, F. Boukouvala, and C. Laird, "Physics-informed machine learning with optimization-based guarantees: Applications to AC power flow," *International Journal of Electrical Power & Energy Systems*, vol. 157, p. 109741, 2024.

[3] B. Donon, R. Clément, B. Donnot, A. Marot, I. Guyon, and M. Schoenauer, "Neural networks for power flow: Graph neural solver," *Electric Power Systems Research*, vol. 189, p. 106547, 2020.

[4] N. Lin, S. Orfanoudakis, N. O. Cardenas, J. S. Giraldo, and P. P. Vergara, "PowerFlowNet: Power flow approximation using message passing graph neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 160, p. 110112, 2024.

[5] L. Piloto, S. Liguori, S. Madjiheurem, M. Zgubic, S. Lovett, H. Tomlinson, S. Elster, C. Apps, and S. Witherspoon, "CANOS: A fast and scalable neural AC-OPF solver robust to N-1 perturbations," 2024. https://arxiv.org/abs/2403.17660.

[6] T. Joswig-Jones, K. Baker, and A. S. Zamzam, "OPF-Learn: An open-source framework for creating representative AC optimal power flow datasets," in *2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2022.

[7] S. Lovett, M. Zgubic, S. Liguori, S. Madjiheurem, H. Tomlinson, S. Elster, C. Apps, S. Witherspoon, and L. Piloto, "OPFData: Large-scale datasets for ac optimal power flow with topological perturbations," 2024. https://arxiv.org/abs/2406.07234.

[8] A. K. Rivera, A. Bhagavathula, A. Carbonero, and P. L. Donti, "PFΔ: A benchmark dataset for power flow under load, generation, and topology variations," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[9] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014.

[10] R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review," 2019.

[11] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, C. Josz, R. Korab, B. Lesieutre, J. Maeght, T. W. K. Mak, D. K. Molzahn, T. J. Overbye, P. Panciatici, B. Park, J. Snodgrass, A. Tbaileh, P. V. Hentenryck, and R. Zimmerman, "The power grid library for benchmarking AC optimal power flow algorithms," 2021.

[12] Y. Chen, Y. Tan, and D. Deka, "Is machine learning in power systems vulnerable?," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2018.

[13] M. H. Dinh, F. Fioretto, M. Mohammadian, and K. Baker, "An analysis of the reliability of AC optimal power flow deep learning proxies," in *2023 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT-LA)*, pp. 170–174, 2023.

[14] S. Chevalier and W. A. Wheeler, "Identifying the smallest adversarial load perturbation that renders dc-opf infeasible," *IEEE Transactions on Power Systems*, pp. 1–12, 2026.

[15] S. Chevalier, D. Starkenburg, R. Parker, and N. Rhodes, "Maximal load shedding verification for neural network models of ac line switching," 2025. https://arxiv.org/abs/2510.23806.

[16] B. Giraud, R. Nellikath, J. Vorwerk, M. Alowaifeer, and S. Chatzivasileiadis, "Neural networks for AC optimal power flow: Improving worst-case guarantees during training," 2025.

[17] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," in *International Conference on Learning Representations*, 2021.

[18] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, and J. P. Vielma, "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization," *Mathematical Programming Computation*, vol. 15, no. 3, pp. 581–589, 2023.

[19] R. B. Parker, B. L. Nicholson, J. D. Siirola, and L. T. Biegler, "Applications of the Dulmage-Mendelsohn decomposition for debugging nonlinear optimization problems," *Computers & Chemical Engineering*, vol. 178, p. 108383, 2023.

[20] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8, June 2018.

[21] O. Dowson, R. B. Parker, and R. Bent, "MathOptAI.jl: Embed trained machine-learning predictors into JuMP models," *INFORMS Journal on Computing*, 2026. *To appear.*

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[23] R. B. Parker, O. Dowson, N. LoGiudice, M. J. Garcia, and R. Bent, "Nonlinear optimization with GPU-accelerated neural network constraints," in *NeurIPS Workshop on GPU-Accelerated and Scalable Optimization*, 2025.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[25] I. S. Duff, "MA57—a code for the solution of sparse symmetric definite and indefinite systems," vol. 30, no. 2, 2004.

[26] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 630–637, Apr. 2020.

[27] N. Rhodes, "PowerPlots.jl: An open source power grid visualization and data analysis framework for academic research," 2025. https://arxiv.org/abs/2510.05063.

I. Review #51A

A. Paper summary

The paper proposes a methodology to generate adversarial inputs for a graph neural network (GNN) model that predicts solutions to the AC power flow problem. The authors formulate two optimization problems that search for input perturbations that maximize the discrepancy between the neural network predictions and the solutions of the AC power flow equations, or that cause the neural network to predict feasible operating points that are actually infeasible. The approach is demonstrated on a GNN trained on a 14-bus system, showing that small perturbations in a limited number of inputs can lead to significant prediction errors, thereby highlighting potential vulnerabilities of machine-learning-based surrogate models used in power system analysis.

B. Comments for authors

The paper addresses an interesting and timely topic, namely the robustness of machine-learning-based surrogate models for power system analysis. As neural network models are increasingly proposed for tasks such as AC power flow approximation, understanding their potential vulnerabilities is an important research direction. The manuscript is clearly written, well structured, and easy to follow, and the proposed methodology is described in a transparent manner.

The core idea of generating adversarial inputs through optimization is well motivated. The two proposed formulations provide a useful way to identify operating points where the neural network predictions significantly deviate from the actual AC power flow solutions, or where the neural network predicts feasible operating points that are actually infeasible. The numerical results clearly illustrate that relatively small perturbations of a limited number of inputs can produce large prediction errors, which supports the authors' argument that robustness testing and verification are important when deploying such surrogate models.

One aspect that would benefit from clarification concerns the dataset used to train the neural network. The manuscript states that 48,600 input samples are used for training but does not explain how these samples are generated. This information is important to interpret the significance of the adversarial examples. In particular, it would be useful to understand whether the adversarial inputs identified by the optimization models lie within or far outside the distribution of operating conditions represented in the training dataset. If the adversarial inputs correspond to unrealistic or highly unlikely operating conditions, their practical relevance may be limited. Conversely, if they are close to realistic operating points, the results would highlight a more significant vulnerability of the model. A short discussion of the dataset generation process and the relation between the adversarial inputs and the training data distribution would strengthen the paper.

Additionally, the experimental evaluation is limited to a single 14-bus test system and one specific GNN architecture. While this is reasonable for a proof-of-concept study, it would be interesting to evaluate whether similar adversarial vulnerabilities appear for other GNN architectures proposed in the literature for AC power flow or AC optimal power flow models, and for larger test systems. Such additional experiments are not strictly necessary for the current contribution but could help assess the generality of the proposed methodology.

Overall, the paper presents an interesting idea and provides clear evidence that adversarial inputs can be constructed for neural network models of AC power flow. Clarifying the characteristics of the training dataset and discussing the relationship between adversarial points and the data distribution would further improve the clarity and impact of the work.

C. Summarize your recommendation in one to two sentences

The paper presents an interesting methodology to construct adversarial inputs for neural-network-based AC power flow models and clearly illustrates potential vulnerabilities of these surrogate models. Clarifying the generation and characteristics of the training dataset would further strengthen the contribution.

— *Salvador Pineda*

II. Review #51B

A. Paper summary

This paper proposes two optimization problems to adversarially attack neural network surrogates for power flow, and explores the implications of running the associated attacks on the CANOS-PF graph neural network model trained for the 14-bus system.

B. Comments for authors

Strengths:

- The general idea of studying the adversarial robustness of power system neural network surrogates is indeed an important and worthwhile area of study.
- The two attack formulations the authors propose are reasonable, with (1) shown to be solvable in most instances and (2) shown to be solvable in 2/3 of instances.

- The results are interesting in showing that an input attack margin of 0.04 can achieve output errors of 0.04-0.06 on voltage magnitude, for CANOS-PF on the 14-bus system. The qualitative analysis that the best perturbations require setting voltage magnitudes of PV buses to their lower bounds, indeed provides useful guidance for training models to be robust from the outset.

Weaknesses:

- It is unclear whether the CANOS-PF model is properly trained, which has large implications for the validity of the results. Notably, the results in Table 1 do not seem to make much sense – the adversarial error is much lower than the standard test set error (are the rows switched?). In addition, the authors chose to train only on unperturbed topologies (N) but test on perturbed topologies (N, N-1, N-2); given that the PFΔ paper and Table 1 establish that such a model does not generalize well to N-1 and N-2 topologies, the model attacked is relatively weak, impacting the significance of the results. It might have made more sense to report results on a version of CANOS-PF trained on N, N-1, and N-2 data.

Questions:

- In Figure 2, it is surprising to me that the "min" differences are not closer to zero – this implies there are no circumstances where CANOS-PF's outputs are close to standard power flow outputs. Is this indeed the case, and if so, are the conclusions to be drawn more-so regarding the strength of CANOS-PF or more-so regarding properties of the adversarial attack?
- Section IV.C.: My understanding is that in the literature, attacks tend to be launched on test points, rather than training points the model has already seen. Here, however, the authors choose to attack training points. What is the rationale for this, and what are the implications?

Other notes:

- For optimization problems (1) and (2), please explicitly list decision variables.
- Page 2: What does the word "senses" mean in this context?
- I believe Table A.7 in PFΔ reports test set error, not train set error. As such, while the authors compare their training set errors in Table 1 to the errors in PFΔ Table A.7., I do not believe this comparison is the correct one.
- In this case, the finding seems to be that the size of the attack (0.04) and the size of the resultant error (0.04-0.06) are on a similar order of magnitude. While the authors pose this as proof that a small attack can lead to desired error, in reality, this seems at first glance like somewhat reasonable model behavior.

C. Summarize your recommendation in one to two sentences

While I think this work is well-motivated and promising, I have some doubts about the quality of the CANOS-PF training and some questions regarding some of the design choices made by the authors in executing this study. As such, I am not sure this paper is ready for presentation in its current form.

— Priya Donti

III. REVIEW #51C

A. Paper summary

There is an increasing interest in graph neural networks trained to estimate the AC Power Flow solution. Results so far show that they can perform equally fast to (or faster than) DC Power Flow but with much higher accuracy.

However, for graph neural networks, so far no methods exist to assess their adversarial robustness in a systematic way. This is critical, as power systems are safety critical systems, and any machine learning method we use must have a degree of trustworthiness. Using insights, knowledge and formulations from conventional neural networks, this paper presents methods that can assess the robustness of graph neural networks.

B. Comments for authors

This is an interesting Notes Paper, presenting two innovative formulations to assess the robustness of graph neural networks used to estimate power flow problems.

Formulations that can assess the adversarial robustness of graph neural networks for power systems are very important. First, because power systems are safety critical systems. Second, because there is an increasing interest in using graph neural networks as AC Power Flow surrogates, given their good performance.

The strength of this paper is that it presents two systematic ways, formulating two optimization problems, to assess the robustness of graph neural networks.

The weaknesses of this paper are the following:

- Scalability? At the moment, the solution of the proposed optimization problems take approximately 40 seconds for the 14-bus system. Are the proposed methods scalable?
- The proposed methods can systematically identify adversarial inputs but they offer no guarantees. It depends a lot on the engineer that performs the test to appropriately define the problems (which inputs to vary, which constraints to consider,

etc) to determine varying adversarial inputs. This information is very important for retraining the graph neural networks and improving their performance. However, still, the operators should use them with caution.

Some comments about clarifications that can help improve the paper:

- Formulation (2) is only presented for lower bounds. However, I would suggest that upper bounds are equally interesting as the lower bounds, e.g. for upper bounds for line flow limits. The authors are encouraged to add a sentence mentioning that it is straightforward to define the counterpart of formulation (2) for an upper bound problem.
- Table II is not referred in the text at all. There should be a sentence explaining what Table II presents.
- Table IV and Fig. 3 need further explanation. It is not clear what they present, especially Fig. 3.
- About Table IV, it must be clarified that v6, v3, v2 are voltages of PV buses and are part of the input setpoints x, x0.
- Again for Table IV: In row 1, you mention v6 = 0.96. But what what the initial value in x0? in other words how much have deviated from the setpoint x0?

Some points (more about expression and language) that need to be addressed in the paper:

- Page 1, Second Paragraph, CANOS PF: there should be a reference to it (i.e. a citation), for the unfamiliar reader.
- First sentence of page 4 needs attention – it somehow does not make sense as written.

C. *Summarize your recommendation in one to two sentences*

This is a PowerUp Notes paper, and it is a very interesting contribution with two innovative formulations of optimization problems to assess the robustness of graph neural networks that estimate power flow solutions.

— *Spyros Chatzivasileiadis*

IV. Brief response to reviewers (optional)

Below I briefly quote the reviewers' main comments and provide my responses.

Reviewer 1:

Quote: *The manuscript states that 48,600 input samples are used for training but does not explain how these samples are generated*

The input samples are taken from PFΔ. The dataset object can be generated with the following Python code:

```
import os
from core.datasets.pfdelta_variants import PFDeltaCANOS

dataset = PFDeltaCANOS(
   add_bus_type=True,
   case_name="case14",
   model="CANOS",
   root_dir=os.path.join("data", "pfdelta_data"),
   split="train",
   task=1.1,
)
```

Refer to our reference [8] and the references therein for details.

Quote: *It would be useful to understand whether the adversarial inputs... lie within or far outside the distribution of operating conditions represented in the training dataset*

I agree that this would be useful. Such an analysis is omitted for brevity. All I can say is that the adversarial points are within the bounds on the input data, a region which should be sampled thoroughly by the training data. If it is not, this paper should be read as more a critique of the training data than the model.

Quote: *It would be interesting to evaluate whether similar adversarial vulnerabilities appear for GNN architectures... and for larger test systems.*

I agree. It seems like it should be harder to train a NN for larger power networks, so I expect errors to persist. However, I don't know whether these adversarial problems will continue to converge quickly and reliably.

Reviewer 2:

Quote: *It is unclear whether the CANOS-PF model is properly trained*

There was a bug in the CANOS loss function, which affected training. See the first item in the summary of changes.

Quote: *Adversarial error is much lower than the standard test set error*

This is because this test set contains topology changes that are omitted in the training data. This is confusing, so test error is omitted.

Quote: *It might have made more sense to report results on a version of CANOS-PF trained on N, N-1, and N-2 data*

This would be interesting and useful, but I believe a "base-case-only" model should still be robust to perturbations in continuous inputs.

Quote: *It is surprising to me that the "min" differences are not closer to zero*

This is the minimum *signed* error, corresponding to a variable with a lower prediction than the power flow solution.

Quote: *The authors choose to attack training points. What is the rationale for this?*

I wanted to attack points where, presumably, the model is accurate. Given the high losses for test points, I thought these would make bad candidates to demonstrate an attack.

Quote: *I believe Table A.7 in PFΔ reports test set error, not train set error... I do not believe this comparison is the correct one.*

I am not sure whether Table A.7 reports train or test errors. None of the tables in the PFΔ paper are a perfect comparison to evaluate our training loss. Table A.7 presents loss for CANOS trained to predict optimal power flow, while Tables A.8-A.12 appear to use test data containing N-1 and N-2 data I did not train on. I have updated the paper to indicate that these are not intended to be one-to-one comparisons.

Reviewer 3:

Quote: *Are the proposed methods scalable?*

This is an open question. From the perspective of function evaluation and linear algebra, there is no reason this should not scale to networks of around 10k buses. Beyond this, the dense Jacobian or Hessian submatrix due to the NN constraints may become a bottleneck. However, convergence reliability may become an issue before this point.

Quote: *[The proposed methods] offer no guarantees [and] depend a lot on the engineer that performs the test to appropriately define the problems.*

I agree. Two comments. One weakness of this method (due to the local nature of the optimization algorithm) is that non-convergence of our method does not guarantee that such an adversarial point does not exist. Second, our constrained-error problem requires the identification of bus(es) and bound(s) to target. However, the "most vulnerable variables" (again without guarantees) can be identified by solving a single problem that maximizes total error and looking at the buses implicated.

V. Brief summary of changes made to the final paper

1) A bug was discovered in the PFΔ CANOS loss function implementation. This impacts the trained CANOS-PF model weights, which impact all the results. I have re-run the analyses with a newly trained version of the model. The biggest difference is an improvement in the training MSE loss obtained. Convergence results and exact adversarial perturbations are slightly different, but conclusions do not change.
2) Removed the last paragraph of Section IV-B.
3) Moved Table II (in the original) to a new subsection, IV-D, where computational details are discussed. Some of this subsection was moved from the conclusion.
4) Clarified that Problem 2 can easily be defined for different combinations of lower or upper bounds.
5) I re-ran the training and optimization solves on a machine with GPUs and made a note of this in Section IV.
6) Clarified the description of Table III, including that the variables shown are included in x as coordinates.
7) Replaced the histogram of perturbation 1-norms with a histogram of 0-norms.
8) Moved the citation for CANOS and PFΔ to the first time these are mentioned in paragraph two of the introduction.
9) Added a link to the GitHub repository that contains code to produce the results.
10) Removed the test loss from Table I. This test set contains N-1 data that the NN was not trained on, leading to confusingly high loss.
11) Updated the description of Table I to indicate that loss tables reported by [8] are not intended to be a one-to-one comparison.