Generalized Data-driven Power Flow Linearization: Methodologies, Evaluations, and Challenges

Response Letter

Mengshuo Jia[‡], Gabriela Hug[‡], Ning Zhang[⋆], Zhaojian Wang[⊕], Yi Wang[⋄], Chongqing Kang[⋆]

[‡]*Power Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland*

[⋆]*Department of Electrical Engineering, Tsinghua University, 100084 Beijing, China*

[⊕]*Department of Automation, Shanghai Jiao Tong University, 200240 Shanghai, China*

[⋄]*Department of Electrical and Electronic Engineering, the University of Hong Kong, 999077 Hong Kong, China*

In order to clearly distinguish the contents in this response letter, the comments of the editor and reviewers are provided in **blue**, the point-to-point responses are given in **black**, and the revised portions of the manuscript included here and in the revised paper are both in **red**.

Please also note that, the dotted-line hyperlinks in this response letter (e.g., **Response to Comment 1.1**) are designed as clickable links for direct navigation to the corresponding responses. However, the submission system's conversion process may disable these interactive features. For convenience, an identical version of this letter with active navigation is available by clicking here.

Response to the Editors

General Comment from the Editor-in-Chief to the Authors: There is consensus among the editorial committee that the two papers, TPWRS-01610-2024 and TPWRS-01611-2024, should be combined into a single paper, the length of which is more consistent with the novelty of the paper, i.e., significantly shorter than the 20 pages currently devoted to the material. This guidance is further reinforced by the existence of a third paper, TPWRS-00059-2025 "Daline: An Open-Source Data-driven Power Flow Linearization Toolbox for Research and Education".

General Comment from the Editor to the Authors: The two-part papers have merit; however none of the reviewers scored the innovation level as high and almost none of the reviewers scored the technical content as high. Some reviewers also recommended rejection. There are several areas where the papers can be improved to ultimately meet the requirements for publication. The topic is generally not new and there is a considerable amount of similar work. If the authors choose to submit a revision, then they should clearly explain the novelty in these papers. Furthermore, some reviewers believe that the paper in its current form does not offer any in-depth insight into these various surveyed methods. The interconnections between different optimization problems should be clearly explained to avoid having a fractured discussion. The relationships between various assumptions need deeper exploration. All in all, the paper, specially the first part on theory, is far from ready. Nevertheless, I would like to recommend an opportunity to submit a major revision for another assessment. I also suggest that the authors consider combining the two papers into a single paper with some in-depth contributions. It might have a higher chance to be ultimately accepted compared to a two-part paper where neither part ends up meeting the requirements for publication.

With regards to Part II, there are again some review comments that need to be addressed that are specific to this part, such as the comments from Reviewers 2 and 3 in Part II. In particular, Reviewer 3 has provided some specific suggestions to combine Part I and Part II. Indeed, my overall assessment is that Part II is more valuable and it could be the primary focus for the combined paper. Again, I think the chance of the paper being ultimately accepted is higher if it is a single but stronger paper, as opposed to having two separate but individually weak papers.

Response to the General Comment: We appreciate the editors' thoughtful guidance and the opportunity to revise and strengthen our manuscript. We also gratefully acknowledge the Editor-in-Chief's support in granting an extended page limit of 16 pages. In response, we have made substantial revisions to address all concerns raised by both editors and reviewers. **As recommended, we have used Part II as the foundation and merged both parts into a single, cohesive manuscript.** Accordingly, the theoretical content has been condensed, evaluations thoroughly refined, the discussion on future challenges shortened, and the Abstract, Introduction, and Conclusion fully rewritten.

On the basis of the revised manuscript, we would like to clarify the **novelty** of our work:

- As a review paper, our objective is not to propose new methodologies, but to provide a comprehensive, tutorial-style contribution with systematic theoretical and numerical evaluations of existing work. Naturally, the topic itself is well-established with a considerable amount of works, as this is essential for meaningful and stable review contributions. While numerous methodological studies on data-driven power flow linearization (DPFL) exist, **there is a clear absence of systematic reviews**, especially those offering large-scale, unified evaluations.

- To the best of our knowledge, **this is the first comprehensive review of DPFL methods, covering 45 methods and including extensive numerical evaluations across 22 test cases**. No comparable work exists in the literature. The only prior overview is a 5-page preliminary summary by our team [1], which only offered a brief, narrative overview without the depth or breadth presented here. Also, the earlier work did not include recent developments or any numerical analysis.
- While several preprints of DPFL review exist, **they are all authored by us**, and none are under peer review or officially published. These preprints simply document our iterative efforts to refine and enhance the systematic review process, culminating in this official and sole journal submission.
- Now, of the four key contributions outlined in the revised manuscript, **two emphasize our unique evaluation work**: comprehensive applicability assessments and numerical tests of 45 methods. We believe this clearly demonstrates the significance and novelty of our contribution within the context of the review literature.

Meanwhile, to systematically address all concerns, we made detailed point-by-point revisions to each reviewer's comment. Below is a summary of major improvements across several key dimensions:

To strengthen the focus and significance of the manuscript, we have:

1) Merged the theoretical and simulation components into a unified, coherent paper to improve the significance of the work, while reflecting the interdependence between concepts and results; see **Response to Comment 3.5** for details.
2) Enhanced consistency between theoretical discussions and numerical findings across all sections, with the addition of a comparative table summarizing all methods' capabilities and limitations; for details, refer to **Response to Comment 1.3** and **Response to Comment 2.1**.
3) Updated the title to better reflect the study's scope as a benchmarking and evaluation work of generalized data-driven linearization methods; see **Response to Comment 1.1** and **Response to Comment 3.1**.
4) Expanded the Conclusion to include practical implementation perspectives; see **Response to Comment 2.2**.

To expand the evaluation scope, depth, and clarity, we have:

1) Integrated four additional recent studies, jointly covering theoretical advancements, capability/limitation analyses, applicability assessments, numerical evaluations, and future research insights; see **Response to Comment 3.4**.
2) Included new distribution test cases to assess performance across transmission and distribution systems; see **Response to Comment 2.8**.
3) Conducted new experiments on noise robustness, with discussions on real-world sensor noise levels; see **Response to Comment 4.4**.
4) Revised noise scenario settings to reflect practical measurement conditions, and updated all corresponding results; see **Response to Comment 4.9**.
5) Added the 1888-bus test case to further evaluate the scalability of the methods; see **Response to Comment 3.9**.
6) Expanded the theoretical discussion to clarify interconnections among optimization-based approaches; see **Response to Comment 1.2**.
7) Included a detailed discussion on localized training strategies, focusing on computational efficiency; see

Response to Comment 4.6.

8) Clarified the validity and scope of Models 1, 2, and 3, with examples of mapping functions; for details, please refer to **Response to Comment 4.3** and **Response to Comment 4.5**.
9) Conducted an analysis of trade-offs between physics-informed and data-driven methods; details are given in **Response to Comment 4.7**.
10) Clarified potentially conflicting limitations; see **Response to Comment 1.4**.

To enhance reproducibility and transparency, we have:

1) Released all evaluated methods in our open-source toolbox DALINE, enabling full reproducibility of results; see **Response to Comment 3.6** for more details.
2) Provided comprehensive documentation of evaluation metrics and improved transparency in experimental setups, including supplementary documentation; see **Response to Comment 3.7** and **Response to Comment 4.11**.

For further details or revisions to any other concerns, please refer to the following point-by-point responses.

Response to Reviewer 1

General Comments to the Author: Overall, this two-part tutorial paper presents a comprehensive overview of data-driven approaches to power flow linearization. The work is commendable, but there are several areas where improvements could be made to enhance clarity and depth. In conclusion, while the paper provides valuable insights into data-driven power flow linearization, addressing these points would significantly improve its clarity and impact.

Response to the General Comment: We thank the reviewer for the recognition and insightful suggestions. The reviewer's comments have been instrumental in refining our work, offering critical perspectives that have significantly improved the clarity, structure, and depth of the manuscript. In response, we have made several substantial revisions, including: **(1) merging the previously separate theory and simulation parts into a unified, coherent manuscript; (2) enhancing the alignment between theoretical discussions and numerical results; (3) expanding the theoretical framework to highlight interconnections among optimization-based approaches; (4) updating the paper title to reflect the inclusion of generalized linear models.** For the specific revisions made in response to each comment, please refer to:

- **Response to Comment 1.1** for the revised title and scope clarification, reflecting the inclusion of generalized linear models and the focus on evaluation rather than theory development.
- **Response to Comment 1.2** for a new subsection that explores interconnections among optimization-based methods, linking objective functions and optimization strategies.
- **Response to Comment 1.3** for the full integration of theoretical and experimental parts, the addition of a summary table, and improved correspondence between claims and results.
- **Response to Comment 1.4** for the clarification of potentially conflicting limitations (i.e., L1 vs. L6).
- **Response to Comment 1.5** for corrections of the identified grammatical issues and comprehensive proofreading to enhance language.

The point-by-point responses are given below.

Comment 1.1: Definition and Scope: The paper considers a very broad definition of linear models, including classical linear models, piecewise linear, and high-dimensional space-mapped (through kernel) linear models. This deviates from the classical definition of linear models. The authors also emphasize that Model 1 is treated as a default unless otherwise specified. It might be more appropriate to redefine the scope and title of the paper to something like "Data-driven Power Flow Estimation" to better reflect the content and approach.

Response to Comment 1.1: As correctly noted by the reviewer, the paper indeed adopts a broad interpretation of "linear models," encompassing classical linearization (Model 1), piecewise-linear formulations (Model 2), and space-mapped but explainable models (Model 3). To better reflect this broader scope while maintaining consistency with terminology used in the existing literature, we have revised the title to:

"Generalized Data-driven Power Flow Linearization: Methodologies, Evaluations, and Challenges"

Accordingly, we have added explanatory text throughout the manuscript to align with the revised title:

Abstract

Generalized data-driven power flow linearization (DPFL) has gained increased attention with the rise of data-centric methodologies, enabling the identification of linear, piecewise linear, and space-mapped power flow models with high accuracy.

Section I. Introduction

... Up to this point, a large number of generalized DPFL approaches have been developed, encompassing not only classical linear models, but also extending to piecewise linear and space-mapped models...

... Different forms of generalized linear models have been proposed in DPFL studies, denoted as follows...

... We also use "DPFL" to denote "generalized DPFL" where the context is clear....

Please also note that in the title we have replaced "Theory and Simulation" with "Methodologies, Evaluations, and Challenges." This change was made to more accurately describe the content of the paper, which focuses on a systematic review of methodologies, large-scale comparative evaluations, and the discussion of future research challenges, rather than proposing new theoretical developments. This change also reflects a similar concern raised by another reviewer, as detailed in ***Response to Comment 3.1***.

Comment 1.2: Theoretical Framework: The theoretical section covers various methods and analyzes their strengths and weaknesses effectively. However, it could be further strengthened by exploring the interconnections between different optimization problems. These connections can be illustrated through the selection of objective functions (e.g., Frobenius norm, Spectral norm, and other norms) and optimization methods such as Chance Constrained Optimization, Robust Optimization, and Distributionally Robust Optimization (DRO). Highlighting these relationships would provide a more cohesive theoretical framework.

Response to Comment 1.2: In the revised manuscript, we have enriched the theoretical discussion by explicitly drawing the connections (and also distinctions) among different optimization schemes, as detailed in the new subsection added at the end of Section III:

Section III.E. Relationships between Programming Methods

There are interconnections but also distinctions among the optimization-based DPFL methods, influencing how these methods balance accuracy, robustness, and outlier detection under different assumptions. (i) **Objective functions**: All formulations involve norm-based residuals. The linearly constrained and distributionally robust methods explicitly minimize residual norms (e.g., ℓ_2 or Frobenius norm) between predicted and true values. The mixed-integer formulation, by contrast, implicitly minimizes the ℓ_1 norm of residuals, but only over the subset of samples not marked as outliers. In the chance-constrained formulation, the objective minimizes $\|\boldsymbol{\beta}\|_F^2/2$ as a regularization term to promote solution stability, since residuals (e.g., in ℓ_1 norm) are already bounded probabilistically via the chance constraints. (ii) **Optimization frameworks**: The distributionally robust formulation generalizes the chance-constrained model by replacing the known data distribution with an ambiguity set, thereby optimizing performance under worst-case distributional shifts. The mixed-integer formulation, meanwhile, excludes outliers using binary variables under a fixed budget, which corresponds to robust residual minimization under combinatorial uncertainty.

Comment 1.3: Experimental Results: There should be a stronger correspondence and comparison between the experimental results and the theoretical discussions. This alignment would help in validating the theoretical claims and providing practical insights. Currently, the two parts seem somewhat disconnected.

Response to Comment 1.3: To strengthen the correspondence between theoretical discussions and experimental results, we have made the following key revisions:

1) We have merged the previously separate theory and simulation papers into a single, coherent manuscript. As a result, the theoretical content, simulation results, as well as the Abstract, Introduction, and Conclusion have been updated to better align both theoretical analysis and numerical evaluation.
2) We have added a comprehensive table summarizing the capabilities and limitations of all reviewed methods. This table serves to highlight the theoretical claims and provide a clear link to the corresponding experimental observations.
3) We have aligned the theoretical discussions with evaluation results across the manuscript. This includes linking the aforementioned summary table with both the applicability evaluations in Section VI and the numerical results in Section VII.

Given the extensive revisions made throughout the manuscript, we have not duplicated all changes here. Instead, we only present the newly added table below as an illustrative example. For the full integration and alignment of the theoretical and experimental content, we kindly invite the reviewer to refer to the revised manuscript.

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
[2]	Ordinary Least Squares	Easy to implement with a clear, interpretable solution.	L1: affected by multicollinearity; L2: outlier-sensitive; L3: assumes heteroskedasticity; L4: costly updates; L5: ignores non-linearity; L6: noise-sensitive.
[3], [4]	Least Squares with COD/SVD	Handles multicollinearity; faster due to reduced triangular factors.	Zero singular values may still hinder inversion; L2 to L6.
[5], [6]	Least Squares with Huber Loss	More tolerant to outliers; alleviates multicollinearity impact.	No closed-form; computationally heavy with hyperparameters tuning; no guarantee on noise efficacy; L3; L4; L5.
[7]	Generalized Least Squares	Accounts for heteroskedasticity & autocorrelation when covariance of residuals known.	True covariance hard to obtain; needs strong assumptions; L1; L2; L4 to L6.
[8]	Total Least Squares	Consider predictors' observational noise as well.	Assumes equal noise variance from various devices; rank-deficiency unresolved; L3 to L5.
[8]	Weighted Total Least Squares	Allows unequal noise treatment for the noise of both predictors and responses.	Non-convex; heuristic approximations; computationally heavy; may reach local optima; L3 to L5.
[9]	Recursive Least Squares	Fast incremental updates; adapts to non-stationarity.	L1 to L3; L6.
[10]	Ordinary Partial Least Squares	Designed to tackle multicollinearity.	Needs latent-dimension tuning; L2 to L6.
[11]	Weighted Recursive Partial Least Squares	Efficient online update; handles multicollinearity.	L2; L3; L5; L6
[12]	Ordinary Ridge Regression	Simple; handles multicollinearity; interpretable.	Penalty tuning expensive; L2 to L6.
[13]	Locally Weighted Ridge Regression	Alleviates non-linearity and multicollinearity with simple closed form.	Requires tuning of more hyperparameters; L2 to L4; L6.
[14]	Clustering-based Ridge Regression	Improves fit to nonlinear patterns; handles multicollinearity.	Many hyperparameters; possible bias and underdetermined clusters; complexify integration into applications; L2 to L4; L6.
[15]	Ordinary Support Vector Regression	Naturally regularized; tolerant to minor outliers; alleviates multicollinearity impact.	Computationally heavy for large systems; hyperparameters tuning expensive; L3 to L5.
[16], [17]	Kernel Support Vector Regression	Alleviates non-linearity while retaining the support vector regression's strengths.	Kernels may not always create a more linear space; high computation with hyperparameters tuning; L3; L4.
[8], [18]	Linearly Constrained Programming (Bound constrained; Structure constrained; Coupling constrained)	Remains convex; leverages physical insight; reduces multicollinearity impact.	May omit valuable predictors; sensitive to scaling; accuracy depends on physical fidelity; limited response set; L2; L4 to L6.
[19]	Chance-constrained Programming	Reduces hyperparameters; clear statistical meaning; robust to small outliers; less multicollinearity risk.	Allows large errors; feasibility sensitive to confidence; big-M method increases problem's scale; computationally heavy for large systems; L3 to L5.
[20]	Distributionally Robust Chance-constrained Programming	Explicit worst-case error control; general robust framework; multicollinearity tolerant.	Needs a preset operating point; computationally heavy for large systems; L3 to L5.
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
[22]	Variable Bundle Strategy	Handles changing bus types dynamically; increases model flexibility.	May cause large linearization errors if key matrices become near-singular.
[2], [12], [14], [15], [18], [19]	Voltage Squaring	Easy to implement; commonly improves model linearity; widely recognized.	No major limitations.
[12]	Voltage-Angle Coupling	Produces interpretable linear models; introduces accurate physics.	May greatly increase the variable number with an unknown topology.
[23]	Dimension Lifting	Flexible transformation; alleviates non-linearity.	Hard to apply in application tasks like dispatch; it increases model complexity.
[10], [24]–[26]	Physical Model Integration (Coefficient Optimization; Error Correction)	Improves accuracy when high-fidelity physical models are available; improves data efficiency.	Sensitive to incorrect physical assumptions; optimization-based methods are computationally heavy.

◇The unshaded rows represent training algorithms, while the shaded ones indicate supportive techniques.
†For illustrative numerical examples, please see Section VII-C, where representative failure cases are identified and analyzed in detail.
‡Table III provides a structured applicability assessment across multiple dimensions, aiding the identification of potential failure scenarios.
⋆Although noise may statistically promote full-rankness, L1 and L6 still represent distinct and practically relevant limitations.

Comment 1.4: Assumptions and Relationships: The relationships between various assumptions need deeper exploration. For instance, the full rank requirement mentioned in L1 and the noise issues in L6 seem contradictory. In the presence of significant noise, a matrix can be considered a random matrix, which is almost surely full rank. Discussing these nuances would clarify potential misunderstandings.

Response to Comment 1.4: The potential tension between L1 (full-rank requirements) and L6 (sensitivity to noise) is indeed subtle. While it is true that additive noise can statistically promote full-rankness, especially from a random matrix theory perspective, there is currently no rigorous theoretical guarantee on when this occurs, nor any established threshold for how much noise is needed to resolve rank deficiency meaningfully.

Therefore, although L1 and L6 may appear somewhat contradictory, they represent distinct and practically relevant limitations: L1 highlights the numerical instability caused by multicollinearity, while L6 emphasizes performance degradation due to the statistical impact of measurement noise. Given their relevance in different contexts, we have kept both in Table I. However, we have also added the following footnote to Table I for clarification:

Table I: Capabilities and Limitations of DPFL Approaches

*Although noise may statistically promote full-rankness, L1 and L6 still represent distinct and practically relevant limitations.

Comment 1.5: Language and Presentation: Given the length of the paper, there are a couple of grammatical errors and typos that need attention. The authors should thoroughly polish the manuscript. For example, ”non-invertible” in L1 should be ”invertible,” and the sentence above Eq. (32) lacks a subject. Careful proofreading will enhance the paper’s readability and professionalism.

Response to Comment 1.5: We have corrected the specific errors mentioned by the reviewer. In addition, we have thoroughly proofread the entire manuscript and used grammar-checking tools to further improve the language and presentation.

Response to Reviewer 2

General Comments to the Author: (Part I) This paper presents a comprehensive tutorial on Data-driven Power Flow Linearization (DPFL). It begins by classifying existing training algorithms, providing an analysis of their mathematical models, capabilities, limitations, and generalizability. The second part focuses on categorizing techniques aimed at enhancing the performance of these training algorithms. Quality: The idea of reviewing existing data-driven methods for linear power flow modeling is certainly valuable. The paper's approach resembles review articles on linear regression methods, such as:

1. Filzmoser, Peter, and Klaus Nordhausen. "Robust linear regression for high-dimensional data: An overview." Wiley Interdisciplinary Reviews: Computational Statistics 13, no. 4 (2021): e1524.
2. Su, Xiaogang, Xin Yan, and Chih-Ling Tsai. "Linear regression." Wiley Interdisciplinary Reviews: Computational Statistics 4, no. 3 (2012): 275-294.
3. Ngu, Joyce Chen Yen, Wan Sieng Yeo, Teck Fu Thien, and Jobrun Nandong. "A comprehensive overview of the applications of kernel functions and data-driven models in regression and classification tasks in the context of software sensors." Applied Soft Computing (2024): 111975.

While the paper does not introduce new concepts in linear algebra, linear regression, or linear model parameterization, its focus on the application of these techniques in power systems — particularly for power flow studies — provides valuable insights for power system engineers.

(Part II) This paper conducts an extensive numerical assessment of 40 data-driven power flow linearization (DPFL) methods and 4 traditional physics-driven power flow linearization (PPFL) approaches, evaluating these methods across different test systems. The study examines their applicability, accuracy, and computational performance. The paper is well-written and discusses an interesting topic. However, some improvements are required for it to be considered for publication.

Response to the General Comment: We would like to thank the reviewer for the recognition and insightful, thoughtful feedback. The reviewer's comments have been instrumental in refining our work, providing critical perspectives that significantly enhance the manuscript. In response, we have made substantial revisions, including **(1) incorporating and evaluating recent studies; (2) supplementing additional distribution cases and comparative results; (3) adding a summary table to highlight the capabilities and limitations of all methods; (4) merging the previously separate theory and simulation parts into a unified, coherent manuscript; (5) expanding the review to include practical implementation perspectives**. For detailed revisions corresponding to each comment, please refer to:

- **Response to Comment 2.1** for the addition of a comparative table summarizing all training algorithms and supportive techniques, in terms of their capabilities and limitations.
- **Response to Comment 2.2** for discussion of how our work informs both practical implementation and future research directions.
- **Response to Comment 2.3** for the integration of four additional recent studies, jointly covering new mathematical formulations, capability and limitation analyses, applicability evaluations, numerical performance

comparisons, and future research directions.

- **Response to Comment 2.4** and **Response to Comment 2.6** for thorough proofreading and addressing of typographical/grammatical errors.
- **Response to Comment 2.5** for a better alignment between theory and practice, and the merging of theoretical and experimental parts into a single, unified manuscript.
- **Response to Comment 2.7** for adding explanations of naming conventions for test cases in the experimental section.
- **Response to Comment 2.8** for additional test cases, new comparative analyses, and performance discussions of various methods across transmission and distribution grids.

The point-by-point responses are detailed in the sections below.

Comment 2.1: Including a comparison table that summarizes all training algorithms along with their pros and cons would offer a quick and valuable insight for readers.

Response to Comment 2.1: We have added this summary table to the revised manuscript:

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
[2]	Ordinary Least Squares	Easy to implement with a clear, interpretable solution.	L1: affected by multicollinearity; L2: outlier-sensitive; L3: assumes heteroskedasticity; L4: costly updates; L5: ignores non-linearity; L6: noise-sensitive.
[3], [4]	Least Squares with COD/SVD	Handles multicollinearity; faster due to reduced triangular factors.	Zero singular values may still hinder inversion; L2 to L6.
[5], [6]	Least Squares with Huber Loss	More tolerant to outliers; alleviates multicollinearity impact.	No closed-form; computationally heavy with hyperparameters tuning; no guarantee on noise efficacy; L3; L4; L5.
[7]	Generalized Least Squares	Accounts for heteroskedasticity & autocorrelation when covariance of residuals known.	True covariance hard to obtain; needs strong assumptions; L1; L2; L4 to L6.
[8]	Total Least Squares	Consider predictors' observational noise as well.	Assumes equal noise variance from various devices; rank-deficiency unresolved; L3 to L5.
[8]	Weighted Total Least Squares	Allows unequal noise treatment for the noise of both predictors and responses.	Non-convex; heuristic approximations; computationally heavy; may reach local optima; L3 to L5.
[9]	Recursive Least Squares	Fast incremental updates; adapts to non-stationarity.	L1 to L3; L6.
[10]	Ordinary Partial Least Squares	Designed to tackle multicollinearity.	Needs latent-dimension tuning; L2 to L6.
[11]	Weighted Recursive Partial Least Squares	Efficient online update; handles multicollinearity.	L2; L3; L5; L6
[12]	Ordinary Ridge Regression	Simple; handles multicollinearity; interpretable.	Penalty tuning expensive; L2 to L6.
[13]	Locally Weighted Ridge Regression	Alleviates non-linearity and multicollinearity with simple closed form.	Requires tuning of more hyperparameters; L2 to L4; L6.
[14]	Clustering-based Ridge Regression	Improves fit to nonlinear patterns; handles multicollinearity.	Many hyperparameters; possible bias and underdetermined clusters; complexify integration into applications; L2 to L4; L6.
[15]	Ordinary Support Vector Regression	Naturally regularized; tolerant to minor outliers; alleviates multicollinearity impact.	Computationally heavy for large systems; hyperparameters tuning expensive; L3 to L5.
[16], [17]	Kernel Support Vector Regression	Alleviates non-linearity while retaining the support vector regression's strengths.	Kernels may not always create a more linear space; high computation with hyperparameters tuning; L3; L4.
[8], [18]	Linearly Constrained Programming (Bound constrained; Structure constrained; Coupling constrained)	Remains convex; leverages physical insight; reduces multicollinearity impact.	May omit valuable predictors; sensitive to scaling; accuracy depends on physical fidelity; limited response set; L2; L4 to L6.
[19]	Chance-constrained Programming	Reduces hyperparameters; clear statistical meaning; robust to small outliers; less multicollinearity risk.	Allows large errors; feasibility sensitive to confidence; big-M method increases problem's scale; computationally heavy for large systems; L3 to L5.
[20]	Distributionally Robust Chance-constrained Programming	Explicit worst-case error control; general robust framework; multicollinearity tolerant.	Needs a preset operating point; computationally heavy for large systems; L3 to L5.
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
[22]	Variable Bundle Strategy	Handles changing bus types dynamically; increases model flexibility.	May cause large linearization errors if key matrices become near-singular.
[2], [12], [14], [15], [18], [19]	Voltage Squaring	Easy to implement; commonly improves model linearity; widely recognized.	No major limitations.
[12]	Voltage-Angle Coupling	Produces interpretable linear models; introduces accurate physics.	May greatly increase the variable number with an unknown topology.
[23]	Dimension Lifting	Flexible transformation; alleviates non-linearity.	Hard to apply in application tasks like dispatch; it increases model complexity.
[10], [24]–[26]	Physical Model Integration (Coefficient Optimization; Error Correction)	Improves accuracy when high-fidelity physical models are available; improves data efficiency.	Sensitive to incorrect physical assumptions; optimization-based methods are computationally heavy.

$^{\diamond}$The unshaded rows represent training algorithms, while the shaded ones indicate supportive techniques.
†For illustrative numerical examples, please see Section VII-C, where representative failure cases are identified and analyzed in detail.
‡Table III provides a structured applicability assessment across multiple dimensions, aiding the identification of potential failure scenarios.
*Although noise may statistically promote full-rankness, L1 and L6 still represent distinct and practically relevant limitations.

Comment 2.2: Discussing how the review can inform future research directions or practical implementation in power systems.

Response to Comment 2.2: This work is intended to act as a guiding resource for both researchers and practitioners. Specifically, the insights gained from the systematic benchmarking of 45 DPFL and PPFL methods can inform the selection and application of suitable techniques in real-world power system tasks, such as economic dispatch, market clearing, and unit commitment.

To further support practical implementation, we would like to mention that a parallel open-source project, DALINE [27], has been developed by us based on this work. While not part of the manuscript's contributions per se, DALINE integrates all evaluated methods and provides a user-friendly modeling pipeline, allowing users to reproduce all the methods in this study with minimal effort. The toolbox and accompanying 150-page user manual are publicly accessible at the official website.

In addition, we would like to emphasize that this review also serves to identify and organize a set of critical future research directions. These include challenges such as handling data pollution, addressing suboptimality, improving computational scalability, etc. These directions are summarized and discussed in Section VIII of the manuscript and are meant to inspire and guide ongoing innovation in the development of DPFL methods.

In the revised manuscript, to reflect the importance of this practical perspective, we have added the following paragraph at the end of the **Conclusion** section:

Section IX. Conclusion

... Eventually, nine open challenges are identified based on theoretical gaps and experimental findings, offering a roadmap to advance DPFL methods.

Beyond analysis, this review is intended to serve as a guiding resource for researchers and practitioners applying DPFL in real-world scenarios such as economic dispatch, market clearing, unit commitment, and grid planning, where reliable and interpretable linear models are critical. A parallel open-source toolbox, DALINE [27], has also been developed by us to support practical implementation, enabling users to reproduce all the methods from this work with minimal effort, and supporting the integration of DPFL into power system applications.

Comment 2.3: The review should be expanded to include newly proposed algorithms/techniques and recent advancements in the field, such as [1] G. Yan and Z. Li, "Construction of an Outlier-Immune Data-Driven Power Flow Model for Model-Absent Distribution Systems," in IEEE Transactions on Power Systems, vol. 39, no. 6, pp. 7449-7452, Nov. 2024.

Response to Comment 2.3: To include the recent advancements in the field, we have now incorporated four additional relevant studies from the literature:

1) Yan, Guoan, and Zhengshuo Li. "Construction of an outlier-immune data-driven power flow model for model-absent distribution systems." *IEEE Transactions on Power Systems*.
2) Taheri, Babak, and Daniel K. Molzahn. "Optimizing parameters of the DC power flow." *Electric Power Systems Research* (the published version of the linearization method used in "Improving the Accuracy of DC Optimal

Power Flow Formulations via Parameter Optimization").

3) Talkington, Samuel, Santiago Grijalva, Matthew J. Reno, Joseph A. Azzolini, and Jouni Peppanen. "Localized structure in secondary distribution system voltage sensitivity matrices." *Electric Power Systems Research*.
4) Nowak, Severin, Chen, Yu Christine, and Wang, Liwei. "Distributed measurement-based optimal DER dispatch with estimated sensitivity models." *IEEE Transactions on Smart Grid*.

Among these, the first paper, i.e., the one mentioned by the reviewer, proposes a novel training method for data-driven power flow linearization. We have therefore added its mathematical formulation to the manuscript and incorporated it into our simulation framework, updating all relevant experiments and comparisons accordingly.

For the remaining three works, their methodologies are already covered in our manuscript. For instance, the method in Taheri et al. employs unconstrained optimization to fine-tune coefficients of the DC power flow model, using solution algorithms such as BFGS and L-BFGS, which are built-in methods in MATLAB's `fminunc` solver. This DC-coefficient-tuning idea has already been discussed in Section IV.C; unconstrained-optimization-based approaches are also introduced and evaluated in our manuscript, using the `fminunc` solver and the same solution algorithm. Nonetheless, we have supplemented our discussion to explicitly cite and reflect on these three additional contributions, ensuring thorough and fair coverage.

Regarding the specific revisions: First, we have incorporated the mixed-integer-programming-based method proposed in "Construction of an outlier-immune data-driven power flow model for model-absent distribution systems." This includes its mathematical formulation, capability and limitation analysis, applicability evaluation, numerical performance evaluation, and a discussion of the corresponding results.

Section III.D. Mixed-integer Programming

To address the challenge of outliers in field measurements, a mixed-integer programming (MIP) formulation has been proposed in [21] to construct an outlier-immune (OI) data-driven power flow model. This method treats each data sample as a candidate for outlier exclusion, and formulates a learning problem that automatically suppresses the influence of suspect data, without the need for heuristic identification steps.

Specifically, given N_s samples $(\boldsymbol{y}_i, \boldsymbol{x}_i)_{i=1}^{N_s}$ and a predefined outlier ratio p, [21] uses the following MIP model to identify the linear power flow model:

$$\min_{\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{u},\boldsymbol{d}} \sum_{i=1}^{N_s} d_i$$

$$\text{s.t.} \quad -Mz_i \leq u_i + (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}) \leq Mz_i, \quad \forall i$$

$$-d_i \leq u_i \leq d_i, \quad \sum_{i=1}^{N_s} z_i \leq p \cdot N_s, \quad z_i \in \{0,1\}, \quad \forall i \tag{34}$$

where z_i indicates whether sample i is an outlier, $d_i \geq |u_i|$ bounds the absolute residual, and M is a large constant.

To accelerate computation, [21] proposes a continuous relaxation-rounding algorithm. It relaxes the integer variables to $z_i \in [0,1]$, solves the resulting linear programming problem, and iteratively fixes

$z_i = 1$ if $z_i > \theta^{OI}$ (i.e., a predefined threshold), until $\sum_i z_i > p \cdot N_s$ is satisfied.

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
⋮	⋮	⋮	⋮
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
⋮	⋮	⋮	⋮

TABLE III: Applicability Evaluation of All the Evaluated DPFL and PPFL Approaches

Approach	Goal	Type	Predictor	Response	Multicollinearity	Zero Data	Constant Data	Normalization
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
`OI`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Section VII.D. Numerical Evaluations

All 45 methods were run on the 22 benchmark cases (9-bus to 1888-bus) for diverse evaluation purposes. Fig. 2 shows the run-time of the methods, while Fig. 3 presents a representative accuracy ranking for active branch flows in the 118-bus-L-N case; the remaining rankings appear in the supplementary material [28]. A summary of all the rankings is provided in Figs. 4 and 5.

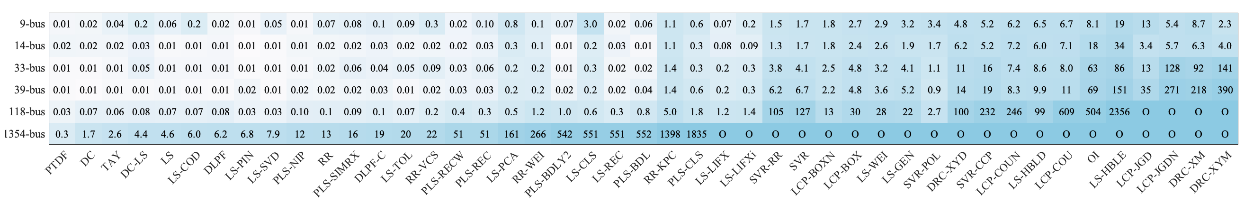

Fig. 2. Computational times of evaluated methods under different test cases (unit: seconds; decimal values for times exceeding 10 seconds are disregarded). The test cases are: 9-bus-S, 14-bus-S, 33-bus-S, 39-bus-S, 118-bus-S, and 1354-bus-S. “O” refers to the OOT-failure. Experiments are carried out on a laptop with an M1 chip and 16 GB of RAM.

Fig. 3. Linearization accuracy ranking of all methods w.r.t. active branch flows, evaluated under the 118-bus-L-N case. The relative error of active branch flow i in sample t is computed by $\epsilon_{it} = |y_{it}^{true} - y_{it}^{method}|/|y_{it}^{true}|$. The mean relative error is thus the average of ϵ_{it} for $\forall i$ and $\forall t$, while the maximal relative error is the maximum of ϵ_{it} for $\forall i$ and $\forall t$. The methods are ranked based on their mean relative errors. `OI`, `TAY`, `LCP_BOX`, `LCP_BOXN`, `PLS_RECW`, `DRC_XM`, and `DRC_XYM` fail and are omitted; see Fig. 4 for their failure types. The other rankings mentioned in Fig. 4 and Fig. 5 are provided in [63]

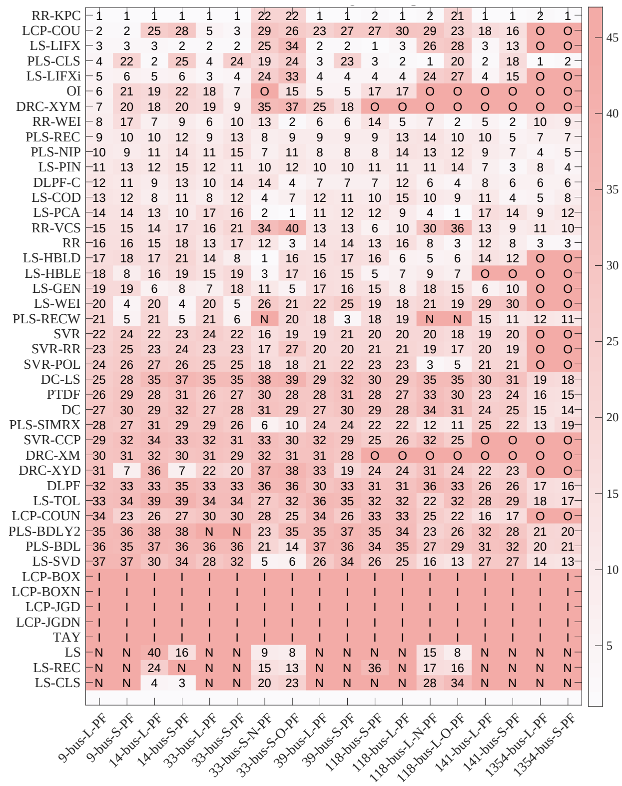

Fig. 4. Rankings of the linearization accuracy of 45 methods w.r.t. active branch flows, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

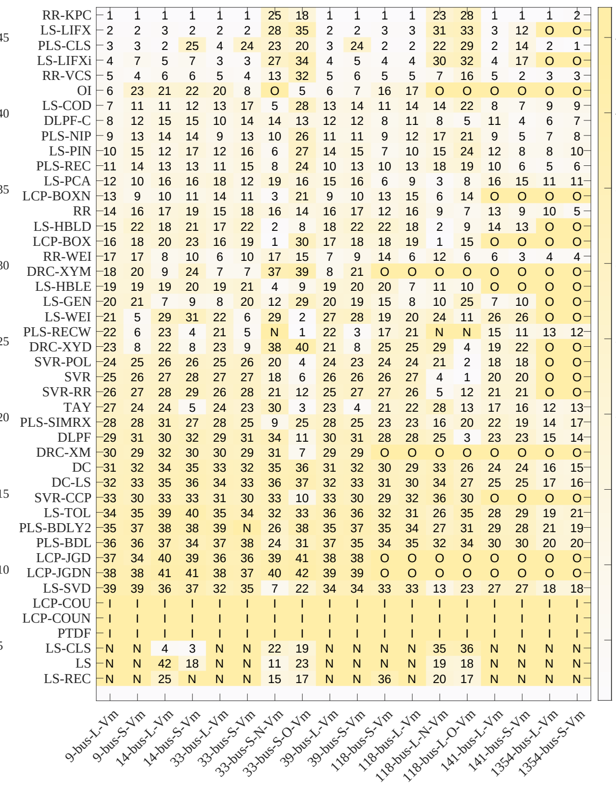

Fig. 5. Rankings of the linearization accuracy of 45 methods w.r.t. voltage magnitudes, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

Section VII.D.3.J. `OI`

As an outlier-robust method, `OI` mitigates the impact of outliers to some extent, as shown in the case of 33-bus-S-O-Vm in Fig. 5. However, the use of the big-M method can lead to failure when numerous constraints couple large coefficients with near-zero decision variables. Such scaling — common in noisy (where most residuals are amplified) or large-scale cases — results in ill-conditioning, which stalls the solver during refinement and prevents completion. This explains the failure of `OI` in the 33-bus-S-N case and in larger systems.

Second, we have extended the discussion regarding the DC-coefficient-optimization by incorporating the reference "Optimizing parameters of the DC power flow," as detailed below:

Section IV.C.1. Coefficient Optimization

Parameters in a PPFL model can be further optimized in a data-driven manner to improve linearization accuracy [25], [26], [29]. In the DC power flow model, for instance, rather than relying solely on coefficients derived from physical line parameters, one can retain the DC model structure while treating the coefficients as trainable parameters. These coefficients can then be optimized using any of the DPFL training algorithms reviewed earlier. For instance, [29] formulates this tuning process as an unconstrained optimization problem, where a loss function measuring the discrepancy between the DC and AC power flow results is minimized (i.e., the same objective formulation principle as in Section III-C). For solutions, gradient-based algorithms, such as quasi-Newton methods implemented in MATLAB's fminunc solver, including BFGS and L-BFGS, can then be applied to optimize the coefficients. Similar ideas are also explored in [25], [26], where either the DC model or the state-independent PPFL model introduced in [30] is employed for data-driven parameter optimization.

Finally, we have incorporated the concept of localized training, as introduced in the two references: "Localized Structure in Secondary Distribution System Voltage Sensitivity Matrices" and "Distributed Measurement-Based Optimal DER Dispatch with Estimated Sensitivity Models," as detailed below:

Section VIII.3. Computational Efficiency

Over 20 methods are computationally intensive, especially for systems with more than 1000 buses. While accurate, many are unsuitable for real-time applications due to their reliance on optimization, nonlinear fitting, or iterative procedures. Improving optimization efficiency, accelerating convergence, enhancing nonlinear modeling, and managing large datasets remain key directions for enabling real-time DPFL. One promising approach is to leverage the grid's physical structure for localized computation. For example, [31] partitions the system into physically defined regions, estimates local models via standard regressions, and coordinates them through distributed optimization, thereby significantly reducing computation without sacrificing accuracy. More recently, [32] pushes localization further by working at the level of secondary load buses, exploiting the natural block structure of voltage sensitivity matrices. Each block is estimated independently using least-squares or ridge regression, achieving substantial speedups with minimal accuracy loss. Though these works do not introduce new DPFL training algorithms, they highlight how physically guided, modular modeling can serve as a scalable alternative to traditional, optimization-heavy approaches. While such methods rely on clear physical separability, they represent a promising direction toward real-time-capable DPFL solutions.

Comment 2.4: Presentation: The manuscript is well-written and easy to follow. However, there are some typos that should be fixed. For example, Page 2, Column 2: The method requires X to be of full column rank such that XTX is "non-invertible" invertible

Response to Comment 2.4: We have addressed the specific issue noted by the reviewer. In addition, we have carefully proofread the entire manuscript to correct additional typographical and grammatical errors.

Comment 2.5: In the introduction, the authors note that existing numerical comparisons in the literature do not fully capture the actual performance of DPFL methods. However, the paper would benefit from a more detailed explanation of why discrepancies arise between theoretical expectations and numerical comparisons. Clarifying the factors contributing to these differences would provide greater context for the need for the comprehensive evaluation presented in this paper. This discussion would strengthen motivation for the study and help readers understand the limitations of prior work more clearly.

Response to Comment 2.5: In the previous version, our intent was to convey that some methods, while theoretically capable of handling specific challenges, may still fail in practice under realistic conditions. These discrepancies are analyzed and explained in detail in the numerical evaluation sections, particularly in Section VII.C (Failure Evaluation) and Section VII.D.3 (Individual Performance of DPFL Methods).

That said, we recognize that the original statement in the Introduction — regarding mismatches between theoretical expectations and numerical performance — may have been overly broad and potentially misleading. It could imply an unintended disconnect between the theoretical analysis and the simulation results. To strengthen the correspondence between theoretical discussions and experimental findings, while retaining the detailed failure explanations provided in the evaluation sections, we have made the following key revisions:

1) We have merged the previously separate theory and simulation papers into a single, coherent manuscript. As a result, the theoretical content, simulation results, as well as the **Abstract**, **Introduction**, and **Conclusion** have been updated to better align both theoretical analysis and numerical evaluation.
2) We have added a comprehensive table summarizing the capabilities and limitations of all reviewed methods, as provided in **Response to Comment 2.1**. This table serves to highlight the theoretical claims and provide a clear link to the corresponding experimental observations.
3) We have aligned the theoretical discussions with evaluation results across the manuscript. This includes linking the aforementioned summary table with both the applicability evaluations in **Section VI** and the numerical results in **Section VII**.

Given the extensive revisions made throughout the manuscript, we have not duplicated all changes here. For the full integration and alignment of the theoretical and experimental content, we kindly invite the reviewer to refer to the revised manuscript.

Comment 2.6: The manuscript contains several minor grammatical and structural inconsistencies that should be addressed to improve clarity and readability. Specifically: On page 3, in the discussion of the RR_VCS method, the phrase "V^2, P, and Q as predictors" is repeated. Consider revising this sentence to eliminate redundancy. Also ensure that all acronyms are defined the first time they appear in the text. For example: The acronym PPFL is not defined in the introduction but is introduced later in Section II. Defining it earlier would improve clarity. On page 6, while terms like INA and OOT are defined, NaN is used without definition. It would be helpful to define NaN similarly for consistency.

Response to Comment 2.6: In the revised manuscript, we have carefully addressed all the issues noted by the reviewer, as detailed below. In addition, we have thoroughly proofread the entire manuscript to ensure consistency and to avoid similar errors throughout the paper.

Section I. Introduction

...

An applicability evaluation of 45 generalized linearization methods—including 37 existing DPFL approaches, four newly developed DPFL methods, and four classic physics-driven power flow linearization (PPFL) algorithms—is conducted, systematically assessing their applicability concerning predictors, responses, multicollinearity, zero data, constant data, and normalization.

...

Section VI.A. Predictor and Response Applicability

...

`RR_VCS` employs the voltage-angle coupling technique, which requires using V^2, P, and Q as predictors while taking V^2, R_{ij}, and C_{ij} for $\forall i, j$ as responses.

...

Section VII.C. Failure Evaluation

...

Figs 4–5 label three distinct failure types, namely **INA** (inapplicable): model cannot produce the requested output; **NaN** (not a number): numerical singularity, mainly caused by multicollinearity; **OOT** (out-of-tolerance): time or memory overruns.

...

Comment 2.7: In the caption of Table III, the various noise and outlier levels considered in the samples are detailed. Similarly, the caption of Figure 1 highlights the small and large fluctuation levels, which correspond to distinct case studies described in Table III. In Table III, 9-bus-S represents the 9-bus system under small fluctuation conditions, while 9-bus-L refers to the same system with large fluctuations. To improve clarity and enhance the reader's understanding, it is recommended to explicitly define these terms and the associated naming conventions within the text.

Response to Comment 2.7: We fully agree that introducing the naming conventions explicitly in the main text can improve clarity and accessibility, even though they were already defined in Table IV in the revised manuscript (formerly Table III). Accordingly, we have added explanatory sentences in the main text where the test cases are first described:

Section VII.A. Experiment Settings

...

Table IV summarises the 22 benchmark cases (from 9-bus to 1888-bus) and the noise/outlier options. Each test case is labeled using a short tag such as "9-bus-S" or "118-bus-L-N," where the suffix "S" and "L" denote small and large fluctuation levels, respectively. Additional suffixes such as "-N" or "-O" indicate the presence of noise or outliers, as detailed in Table IV.

...

Comment 2.8: The paper provides a comprehensive comparison of various DPFL and PPFL methods across different case studies. To further enhance the paper, the authors could provide a comparative analysis that explicitly differentiates the performance of these methods between transmission and distribution case studies. Highlighting which methods perform better for each type of system and explaining why certain approaches may be more effective in distribution grids versus transmission networks, would offer deeper insights and strengthen the discussion.

Response to Comment 2.8: To provide a more specified comparison between transmission and distribution systems, we have made the following key revisions in the manuscript:

1) We have expanded the test case pool to include additional distribution grids, notably the 69-bus and 141-bus radial distribution systems, thereby improving the representation of distribution-level networks in our analysis.
2) We have implemented new evaluations to explicitly compare and discuss the performance of both data-driven and physics-driven methods across transmission and distribution cases.

Please note that the whole manuscript is subject to a 16-page limit set by the Editor-in-Chief. Accordingly, we have aimed to present the content as clearly and concisely as possible. Detailed revisions are provided below:

Section VII.A. Experiment Settings

...

TABLE IV: Settings for the Test Cases

Test Case	Training Samples	Testing Samples	Fluctuation Level	Grid Type	Noise Level◇	Outlier Level◇
9-bus-S	150	100	95% - 105%	Transmission	-	-
9-bus-L	150	100	80% - 120%	Transmission	-	-
14-bus-S	200	150	95% - 105%	Transmission	-	-
14-bus-L	200	150	80% - 120%	Transmission	-	-
33-bus-S	300	200	95% - 105%	Distribution	-	-
33-bus-S-N	300	200	95% - 105%	Distribution	45dB	-
33-bus-S-O	300	200	95% - 105%	Distribution	-	2%
33-bus-L	300	200	80% - 120%	Distribution	-	-
39-bus-S	300	200	95% - 105%	Transmission	-	-
39-bus-L	300	200	80% - 120%	Transmission	-	-
69-bus-S	300	200	95% - 105%	Distribution	-	-
69-bus-L	300	200	80% - 120%	Distribution	-	-
118-bus-S	400	300	95% - 105%	Transmission	-	-
118-bus-L	400	300	80% - 120%	Transmission	-	-
118-bus-L-N	400	300	80% - 120%	Transmission	45dB	-
118-bus-L-O	400	300	80% - 120%	Transmission	-	2%
141-bus-S	400	200	95% - 105%	Distribution	-	-
141-bus-L	400	200	80% - 120%	Distribution	-	-
1354-bus-S	3000	1000	95% - 105%	Transmission	-	-
1354-bus-L	3000	1000	80% - 120%	Transmission	-	-
1888-bus-S	4000	1000	95% - 105%	Transmission	-	-
1888-bus-L	4000	1000	80% - 120%	Transmission	-	-

‡The terminal voltages of generators are fixed during data generation.
◇Noise and outliers are added only to the training data; testing data remains clean. Noise is modeled as white Gaussian at 45 dB [33], with each data point independently perturbed, simulating measurements from separate devices. For outliers, 2% of values in each variable (column) are randomly doubled, representing device-specific errors. Although this rate is small, the combined effect across variables increases the likelihood that a sample (row) contains at least one outlier.

...

Section VII.D.B. Performance across Grid Types

Unlike PPFL methods that often require system-specific adaptations, DPFL approaches are designed to be broadly applicable across different grid types. As shown in Fig. 7, representative DPFL methods exhibit comparable levels of accuracy in both transmission and distribution networks, outperforming PPFL methods in both types. While the results depend on the specific systems tested and their inherent linearity under the given operating conditions, they nonetheless demonstrate the potential of DPFL methods to generalize across diverse grid settings.

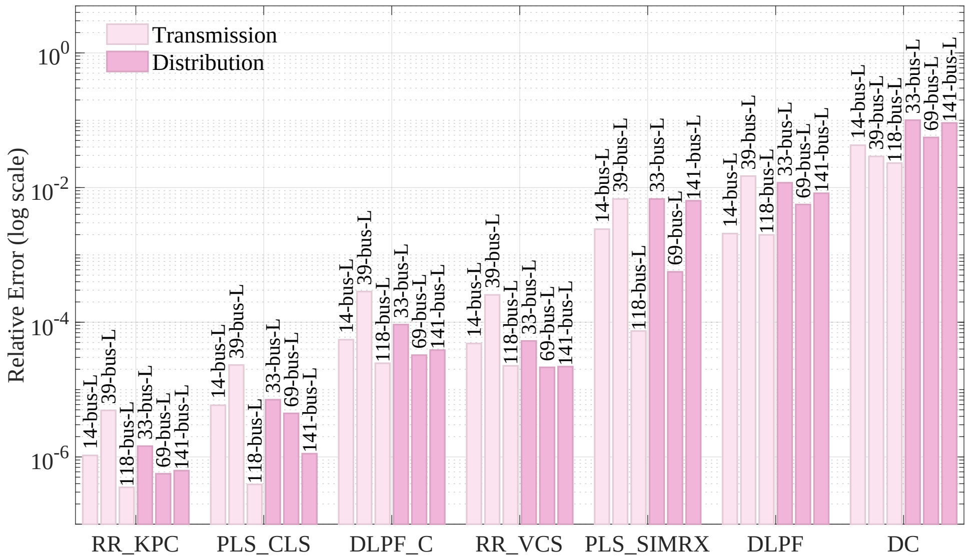

Fig. 7: Relative voltage errors (in average) of representative methods under various transmission and distribution networks.

Response to Reviewer 3

General Comments to the Author: (Part I) Thank you for working on this paper. Below are my observations and comments on the paper. (Part II) This is a continuum to my review for the theory part. The paper runs simulations and classifies their results based on simulation time and linearization accuracy. Overall, the papers have done a commendable job collecting and exploring various linearization techniques for power flow models. However, I found the majority of the literature in the theory paper to be at a very high level, and the experiments section in the simulation paper lacked technical details.

Response to the General Comment: We sincerely appreciate the reviewer's time and effort in evaluating our work. The reviewer's thoughtful and constructive comments have been invaluable in improving the quality, clarity, technical depth, and completeness of our manuscript. In response, we have made several major revisions: **(1) the two original manuscripts have been merged into a unified paper; (2) recent literature has been incorporated and evaluated in both analytical and numerical ways; (3) a large-scale test case has been added; (4) the full set of evaluated methods has been open-sourced**. For specific revisions made in response to each comment, please refer to:

- **Response to Comment 3.1** for the revised title, clarifying that this is a review and benchmarking study rather than a theoretical proposal.
- **Response to Comment 3.2** for a new summary table and accompanying revisions that explicitly highlight the failure conditions of different methods.
- **Response to Comment 3.3** for an expanded discussion and evaluation of algorithmic strategies to address multicollinearity, along with a clarification of the paper's scope.
- **Response to Comment 3.4** for the integration of four additional recent studies, jointly covering new mathematical formulations, capability and limitation analyses, applicability evaluations, numerical performance comparisons, and future research directions.
- **Response to Comment 3.5** for merging the theory and simulation manuscripts into a single, coherent work as recommended.
- **Response to Comment 3.6** for the release of our open-source toolbox DALINE, which includes all evaluated methods and supports complete reproducibility.
- **Response to Comment 3.7** for clarification of the evaluation metrics used in Figs. 2–4 to ensure reproducibility.
- **Response to Comment 3.8** for the addition of a dedicated section on topological variation, identifying it as an important open challenge for future research.
- **Response to Comment 3.9** for the inclusion of the 1888-bus French transmission system to enable performance evaluation on large-scale networks.
- **Response to Comment 3.10** for corrections of typographical and grammatical errors.
- **Response to Comment 3.11** for clarification of the term "predictor and response applicability," explanation of the purpose of Section VI, and improved terminology throughout the manuscript.

Below, we provide the revisions made in response to each comment.

Comment 3.1: The paper is titled: “Data-driven Power Flow Linearization: Theory”. However, the paper is a review paper, not a manuscript where a new theory is developed. So, in that sense, I felt the title is misleading.

Response to Comment 3.1: We agree that the original title could be misleading. Since the paper primarily presents a systematic review of methodologies, comparative evaluations, and future directions, we have revised the title to:

“Generalized Data-driven Power Flow Linearization: Methodologies, Evaluations, and Challenges”

Please note that the term “Generalized” was introduced to reflect the broader scope of linearization considered in this work, including classical, piecewise, and space-mapped but explainable models, which extends beyond the traditional definition. This adjustment also aligns with a concern raised by another reviewer regarding the scope of linear models; please refer to ***Response to Comment 1.1*** for details.

Comment 3.2: Further, most review methods for building linear models are covered at a high level. I could not develop any deep insights into when a certain method may work or fail. Simple examples which show where a certain method would fail may help here.

Response to Comment 3.2: We fully understand that the original manuscript may not have clearly highlighted the key takeaways, especially regarding failure cases, which were previously embedded within detailed mathematical formulations and capability/limitation analyses. To address this, we have made the following improvements:

1) **Added a comprehensive summary table (Table I)** that outlines the capabilities and limitations of each method, explicitly highlighting conditions under which certain methods may fail.
2) **Revised and retained Table III**, despite page constraints. This table provides a structured assessment of each method’s applicability across multiple dimensions—such as predictors, responses, multicollinearity, zero/constant data, and normalization—facilitating the identification of potential failure scenarios.
3) **Included footnotes in Table I** directing readers to two key sources of specific failure examples: (i) Table III, as noted above, and (ii) Section VII-C, which presents detailed numerical examples of failure cases.

The newly added and revised tables are provided on the following two pages.

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
[2]	Ordinary Least Squares	Easy to implement with a clear, interpretable solution.	L1: affected by multicollinearity; L2: outlier-sensitive; L3: assumes heteroskedasticity; L4: costly updates; L5: ignores non-linearity; L6: noise-sensitive.
[3], [4]	Least Squares with COD/SVD	Handles multicollinearity; faster due to reduced triangular factors.	Zero singular values may still hinder inversion; L2 to L6.
[5], [6]	Least Squares with Huber Loss	More tolerant to outliers; alleviates multicollinearity impact.	No closed-form; computationally heavy with hyperparameters tuning; no guarantee on noise efficacy; L3; L4; L5.
[7]	Generalized Least Squares	Accounts for heteroskedasticity & autocorrelation when covariance of residuals known.	True covariance hard to obtain; needs strong assumptions; L1; L2; L4 to L6.
[8]	Total Least Squares	Consider predictors' observational noise as well.	Assumes equal noise variance from various devices; rank-deficiency unresolved; L3 to L5.
[8]	Weighted Total Least Squares	Allows unequal noise treatment for the noise of both predictors and responses.	Non-convex; heuristic approximations; computationally heavy; may reach local optima; L3 to L5.
[9]	Recursive Least Squares	Fast incremental updates; adapts to non-stationarity.	L1 to L3; L6.
[10]	Ordinary Partial Least Squares	Designed to tackle multicollinearity.	Needs latent-dimension tuning; L2 to L6.
[11]	Weighted Recursive Partial Least Squares	Efficient online update; handles multicollinearity.	L2; L3; L5; L6
[12]	Ordinary Ridge Regression	Simple; handles multicollinearity; interpretable.	Penalty tuning expensive; L2 to L6.
[13]	Locally Weighted Ridge Regression	Alleviates non-linearity and multicollinearity with simple closed form.	Requires tuning of more hyperparameters; L2 to L4; L6.
[14]	Clustering-based Ridge Regression	Improves fit to nonlinear patterns; handles multicollinearity.	Many hyperparameters; possible bias and underdetermined clusters; complexify integration into applications; L2 to L4; L6.
[15]	Ordinary Support Vector Regression	Naturally regularized; tolerant to minor outliers; alleviates multicollinearity impact.	Computationally heavy for large systems; hyperparameters tuning expensive; L3 to L5.
[16], [17]	Kernel Support Vector Regression	Alleviates non-linearity while retaining the support vector regression's strengths.	Kernels may not always create a more linear space; high computation with hyperparameters tuning; L3; L4.
[8], [18]	Linearly Constrained Programming (Bound constrained; Structure constrained; Coupling constrained)	Remains convex; leverages physical insight; reduces multicollinearity impact.	May omit valuable predictors; sensitive to scaling; accuracy depends on physical fidelity; limited response set; L2; L4 to L6.
[19]	Chance-constrained Programming	Reduces hyperparameters; clear statistical meaning; robust to small outliers; less multicollinearity risk.	Allows large errors; feasibility sensitive to confidence; big-M method increases problem's scale; computationally heavy for large systems; L3 to L5.
[20]	Distributionally Robust Chance-constrained Programming	Explicit worst-case error control; general robust framework; multicollinearity tolerant.	Needs a preset operating point; computationally heavy for large systems; L3 to L5.
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
[22]	Variable Bundle Strategy	Handles changing bus types dynamically; increases model flexibility.	May cause large linearization errors if key matrices become near-singular.
[2], [12], [14], [15], [18], [19]	Voltage Squaring	Easy to implement; commonly improves model linearity; widely recognized.	No major limitations.
[12]	Voltage-Angle Coupling	Produces interpretable linear models; introduces accurate physics.	May greatly increase the variable number with an unknown topology.
[23]	Dimension Lifting	Flexible transformation; alleviates non-linearity.	Hard to apply in application tasks like dispatch; it increases model complexity.
[10], [24]–[26]	Physical Model Integration (Coefficient Optimization; Error Correction)	Improves accuracy when high-fidelity physical models are available; improves data efficiency.	Sensitive to incorrect physical assumptions; optimization-based methods are computationally heavy.

$^{\diamond}$The unshaded rows represent training algorithms, while the shaded ones indicate supportive techniques.
†For illustrative numerical examples, please see Section VII-C, where representative failure cases are identified and analyzed in detail.
‡Table III provides a structured applicability assessment across multiple dimensions, aiding the identification of potential failure scenarios.
*Although noise may statistically promote full-rankness, L1 and L6 still represent distinct and practically relevant limitations.

TABLE III: Applicability Evaluation of All the Evaluated DPFL and PPFL Approaches

Approach	Goal	Type	Predictor	Response	Multicollinearity	Zero Data	Constant Data	Normalization
`LS`	Build	1	Unconstrained	Unconstrained	×	×	✓	✓
`LS_SVD`	Build	1	Unconstrained	Unconstrained	×	✓	✓	✓
`LS_COD`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_HBLD`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_HBLE`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_TOL`	Build	1	Unconstrained	Unconstrained	×	✓	✓	✓
`LS_CLS`	Build	1	Unconstrained	Unconstrained	×	×	✓	✓
`LS_LIFX`	Build	3	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_LIFXi`	Build	3	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_WEI`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_REC`	Update	1	Unconstrained	Unconstrained	×	×	✓	✓
`PLS_SIMRX`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`PLS_BDL`	Build	1	Fixed: V, P, Q	Unconstrained	✓	×	×	✓
`PLS_BDLY2`	Build	1	Fixed: V, P, Q	Unconstrained	✓	×	×	✓
`PLS_REC`	Update	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`PLS_RECW`	Update	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`RR`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`RR_VCS`	Build	3	Fixed: V^2, P, Q	Fixed: V^2, R_{ij}, C_{ij}	✓	✓	✓	×
`RR_KPC`	Build	2	Unconstrained	Unconstrained	✓	✓	✓	✓
`RR_WEI`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`SVR`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`SVR_CCP`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`SVR_POL`	Build	3	Unconstrained	Unconstrained	✓	✓	✓	✓
`SVR_RR`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LCP_BOX`	Build	1	Fixed: P, Q	Fixed: V, θ	✓	✓	✓	×
`LCP_COU`	Build	1	Fixed: V, θ	Fixed: PF, PT, QF, QT	✓	✓	✓	×
`LCP_JGD`	Build	1	Fixed: P, Q	Fixed: V, θ	✓	✓	✓	×
`DRC_XM`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`DRC_XYM`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`DRC_XYD`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`OI`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`DC_LS`	Build	1	Fixed: P	Fixed: θ	✓	✓	✓	×
`DLPF_C`	Build	1	Fixed: V, θ, P, Q	Fixed: V, θ, PF, QF	✓	✓	✓	×
`LS_PIN`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_PCA`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`LS_GEN`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`PLS_CLS`	Build	2	Unconstrained	Unconstrained	✓	✓	✓	✓
`PLS_NP`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
`DC`	Build	1	Fixed: P	Fixed: θ	✓	✓	✓	×
`PTDF`	Build	1	Fixed: P	Fixed: PF	✓	✓	✓	×
`TAY`	Build	1	Fixed: P, Q	Fixed: V, θ	✓	✓	✓	×
`DLPF`	Build	1	Fixed: V, θ, P, Q	Fixed: V, θ, PF, QF	✓	✓	✓	×

† "Build" refers to constructing a linear model from a static historical dataset of electrical measurements; "Update" denotes refining an existing model by recursively incorporating new data; "Unconstrained" indicates that the method can use any set of predictors or responses without a predefined structure.

Comment 3.3: For instance, a major theme in the paper is that data in power systems can exhibit multicollinearity. This is true as you may have multiple data points that are not linearly independent (or very close to linear dependence). If this is true, many solutions discussed in the paper will not help unless independent data samples are obtained. This is an inherent data issue and requires a data solution, which wasn't discussed in the paper. Put another way, while the algorithms may give a result using algorithms like QR decomposition if most data samples are multicollinear, the quality of the obtained linear model is unlikely to be good.

Response to Comment 3.3: We completely agree that, at the data level, approaches such as decorrelation or acquiring more independent samples can help mitigate multicollinearity.

However, we respectfully suggest that there may have been a misunderstanding regarding the scope and focus of our work. **This paper is not intended to propose new data collection or preprocessing strategies. Instead, it aims to systematically evaluate DPFL algorithms from an algorithmic perspective**, including how well they handle realistic data challenges such as multicollinearity. We seek to assess them objectively, without assuming they will succeed.

To this end, multicollinearity, as one of the core data issues, is explicitly defined, evaluated, and addressed throughout the paper. Algorithmic strategies to mitigate or resolve collinearity, **extending far beyond just QR decomposition**, are systematically reviewed and tested. These include, but are not limited to:

1) The complete-orthogonal-decomposition-based least squares method (i.e., `LS_COD`), which stabilizes training under ill-conditioned predictors by decomposing the inversion process (Sec. II.A.2).
2) Dimension-lifting-based least squares methods (i.e., `LS_LIFX` and `LS_LIFXi`), which address multicollinearity by transforming the original collinear data space into a higher-dimensional feature space (Sec. IV.B.3).
3) Partial least squares (PLS) methods (e.g., `PLS_SIMRX`, `PLS_BDL`, etc.), which extract latent orthogonal components and remove slack bus data to alleviate multicollinearity (Sec. II.B; Sec. IV.A).
4) Ridge regression and its variants (e.g., `RR`, `RR_KPC`, `RR_WEI`), which apply Tikhonov regularization to reduce the impact of near-singular predictor matrices resulting from high correlation (Sec. II.C.1–3).
5) Weight-based regression methods (i.e., `LS_WEI` and `RR_WEI`), which handle multicollinearity by assigning different weights to data samples, thereby breaking the underlying collinear pattern (Sec. II.C.2).
6) Support vector regression approaches (e.g., `SVR`, `SVR_RR`, etc.), which also adopt Tikhonov regularization to reduce the degree of singularity caused by multicollinearity (Sec. II.D).
7) Optimization-based methods (e.g., `LCP`-based and `DRC`-based methods), which alleviate multicollinearity by avoiding direct inversion of the Gram matrix of predictors (Sec. III).
8) Physics–data hybrid approaches (e.g., `DLPF_C`, `DC_LS`, etc.), which bypass the use of collinear predictor data by training on the error between a predefined physics-based linear model and the AC model, thus reducing multicollinearity (Sec. IV.C).

Beyond theoretical discussion, we explicitly assess the robustness of all methods against multicollinearity in Table III (column "Multicollinearity"). Additionally, Section VII.C (Failure Evaluation) analyzes algorithmic failures caused by multicollinearity (i.e., NaN failures). In particular, Section VII.D.1, based on large-scale numerical evidence, concludes that "dataset size does influence (but does not guarantee relief from) multicollinearity." Furthermore, the entire Section VII.D demonstrates that the aforementioned multicollinearity-robust algorithms remain consistently stable across more than 40 test cases. **More importantly, these algorithms demonstrate strong practical performance**, especially in scenarios where traditional methods fail due to severe multicollinearity, as shown in Figs. 3–7.

Overall, since multicollinearity is often unavoidable in real-world applications due to physical coupling in power systems, **addressing this issue from the algorithmic side is both important and practical**, even if it does not qualify as a "data solution" in the strictest sense.

To avoid further misunderstandings, we have clarified the scope of our work in the revised manuscript by explicitly stating the following:

Section I. Introduction → Remark

... (ii) Rather than altering the data structure itself, this work focuses on evaluating the algorithmic resilience of various DPFL methods under diverse data conditions.

Comment 3.4: I also feel there is recent literature in the field that is not discussed within this paper, such as recent work on improving DC-OPF solutions based on parameter optimization by Taheri et al. Also, many ML approaches that are equally applicable here are not discussed, especially with the title focusing on data-driven. Deep-learning is a naive example.

Response to Comment 3.4*:* We fully agree with the reviewer that the paper should be as comprehensive and up-to-date as possible, and we have made every effort to achieve this. However, we would like to clarify that the paper referenced by the reviewer was uploaded to arXiv on October 15, 2024, whereas our manuscript was submitted to the journal on September 22, 2024.

Nevertheless, to enhance the completeness of our review, **we have now incorporated four additional relevant studies** from the literature:

1) Taheri, Babak, and Daniel K. Molzahn. "Optimizing parameters of the DC power flow." *Electric Power Systems Research* (the published version of the linearization method used in "Improving the Accuracy of DC Optimal Power Flow Formulations via Parameter Optimization").
2) Yan, Guoan, and Zhengshuo Li. "Construction of an outlier-immune data-driven power flow model for model-absent distribution systems." *IEEE Transactions on Power Systems*.
3) Talkington, Samuel, Santiago Grijalva, Matthew J. Reno, Joseph A. Azzolini, and Jouni Peppanen. "Localized structure in secondary distribution system voltage sensitivity matrices." *Electric Power Systems Research*.
4) Nowak, Severin, Chen, Yu Christine, and Wang, Liwei. "Distributed measurement-based optimal DER dispatch with estimated sensitivity models." *IEEE Transactions on Smart Grid*.

Among these, the second paper proposes a novel training method for data-driven power flow linearization. We have therefore added its mathematical formulation to the manuscript and incorporated it into our simulation framework, updating all relevant experiments and comparisons accordingly.

For the remaining three works, their methodologies are already covered in our manuscript. For instance, the method in Taheri et al. employs unconstrained optimization to fine-tune coefficients of the DC power flow model, using solution algorithms such as BFGS and L-BFGS, which are built-in methods in MATLAB's `fminunc` solver. This DC-coefficient-tuning idea has already been discussed in Section IV.C; unconstrained-optimization-based approaches are also introduced and evaluated in our manuscript, using the `fminunc` solver and the same solution algorithm. Nonetheless, we have supplemented our discussion to explicitly cite and reflect on these three additional contributions, ensuring thorough and fair coverage.

In addition, regarding machine learning (ML) approaches—specifically deep learning—as suggested by the reviewer, we would like to clarify the following: The goal of power flow linearization is to develop an explainable, transparent, and ideally linear model that accurately approximates the AC power flow equations, while remaining suitable for integration into practical power system applications such as economic dispatch or optimization-based control. While deep learning methods have demonstrated remarkable performance across a wide range of end-to-end tasks, their outputs are typically highly nonlinear and embedded within deep neural networks that lack interpretability and transparency. These characteristics are **fundamentally misaligned with the objectives of power flow linearization**, which explicitly favors analytical tractability. For this reason, **deep-learning-based methods fall outside the scope of this paper and are not included in our evaluation**. Nevertheless, within the broader class of ML-based approaches, we have already included all the methods we have found in our study, such as support vector regression and various regression-based algorithms, which are more compatible with the goals of linearization and retain a higher degree of model interpretability.

Regarding the specific revisions: First, we have incorporated the mixed-integer-programming-based method proposed in “Construction of an outlier-immune data-driven power flow model for model-absent distribution systems.” This includes its mathematical formulation, capability and limitation analysis, applicability evaluation, numerical performance evaluation, and a discussion of the corresponding results.

Section III.D. Mixed-integer Programming

To address the challenge of outliers in field measurements, a mixed-integer programming (MIP) formulation has been proposed in [21] to construct an outlier-immune (OI) data-driven power flow model. This method treats each data sample as a candidate for outlier exclusion, and formulates a learning problem that automatically suppresses the influence of suspect data, without the need for heuristic identification steps.

Specifically, given N_s samples $(\boldsymbol{y}_i, \boldsymbol{x}_i)_{i=1}^{N_s}$ and a predefined outlier ratio p, [21] uses the following MIP model to identify the linear power flow model:

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{u},\boldsymbol{d}} \sum_{i=1}^{N_s} d_i \\ \text{s.t.} \quad & -Mz_i \le u_i + (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}) \le Mz_i, \quad \forall i \\ -d_i \le u_i \le d_i, \quad & \sum_{i=1}^{N_s} z_i \le p \cdot N_s, \quad z_i \in \{0,1\}, \quad \forall i \end{aligned} \tag{34}$$

where z_i indicates whether sample i is considered an outlier, $d_i \ge |u_i|$ bounds the absolute residual, and M is a sufficiently large constant.

To accelerate computation, [21] proposes a continuous relaxation-rounding algorithm. It relaxes the integer variables to $z_i \in [0,1]$, solves the resulting linear programming problem, and iteratively fixes $z_i = 1$ if $z_i > \theta^{OI}$ (i.e., a predefined threshold), until $\sum_i z_i > p \cdot N_s$ is satisfied.

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
⋮	⋮	⋮	⋮
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
⋮	⋮	⋮	⋮

TABLE III: Applicability Evaluation of All the Evaluated DPFL and PPFL Approaches

Approach	Goal	Type	Predictor	Response	Multicollinearity	Zero Data	Constant Data	Normalization
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
`OI`	Build	1	Unconstrained	Unconstrained	✓	✓	✓	✓
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Section VII.D. Numerical Evaluations

All 45 methods were run on the 22 benchmark cases (9-bus to 1888-bus) for diverse evaluation purposes. Fig. 2 shows the run-time of the methods, while Fig. 3 presents a representative accuracy ranking for active branch flows in the 118-bus-L-N case; the remaining rankings appear in the supplementary material [28]. A summary of all the rankings is provided in Figs. 4 and 5.

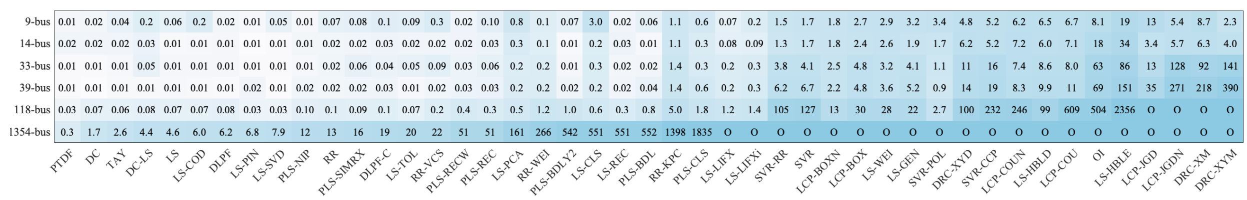

Fig. 2. Computational times of evaluated methods under different test cases (unit: seconds; decimal values for times exceeding 10 seconds are disregarded). The test cases are: 9-bus-S, 14-bus-S, 33-bus-S, 39-bus-S, 118-bus-S, and 1354-bus-S. "O" refers to the OOT-failure. Experiments are carried out on a laptop with an M1 chip and 16 GB of RAM.

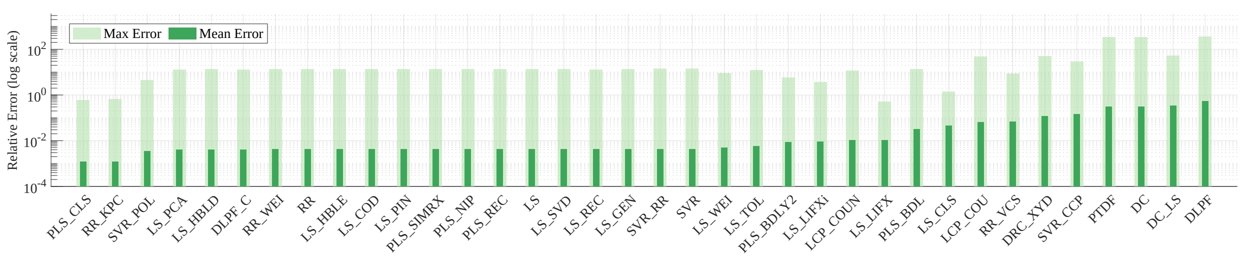

Fig. 3. Linearization accuracy ranking of all methods w.r.t. active branch flows, evaluated under the 118-bus-L-N case. The relative error of active branch flow i in sample t is computed by $\epsilon_{it} = |y_{it}^{true} - y_{it}^{method}|/|y_{it}^{true}|$. The mean relative error is thus the average of ϵ_{it} for $\forall i$ and $\forall t$, while the maximal relative error is the maximum of ϵ_{it} for $\forall i$ and $\forall t$. The methods are ranked based on their mean relative errors. `OI`, `TAY`, `LCP_BOX`, `LCP_BOXN`, `PLS_RECW`, `DRC_XM`, and `DRC_XYM` fail and are omitted; see Fig. 4 for their failure types. The other rankings mentioned in Fig. 4 and Fig. 5 are provided in [63]

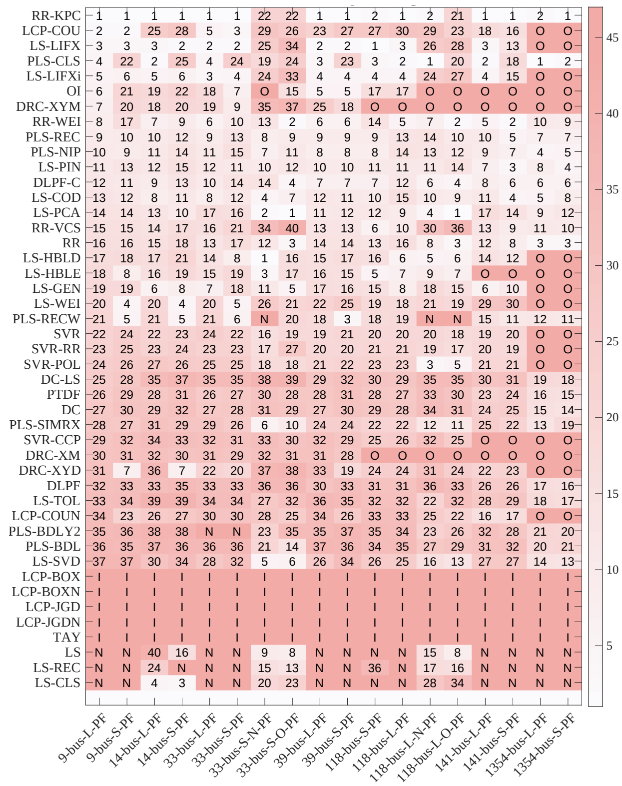

Fig. 4. Rankings of the linearization accuracy of 45 methods w.r.t. active branch flows, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

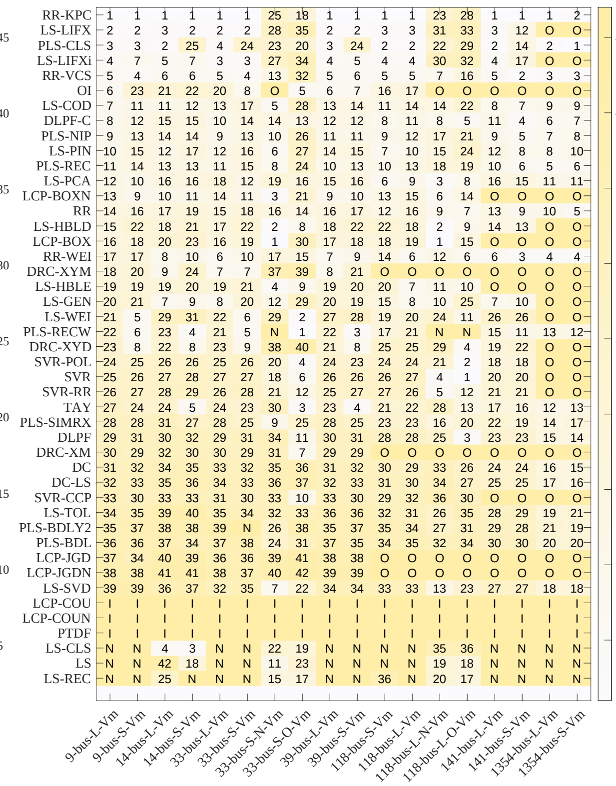

Fig. 5. Rankings of the linearization accuracy of 45 methods w.r.t. voltage magnitudes, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

Section VII.D.3.J. `OI`

As an outlier-robust method, `OI` mitigates the impact of outliers to some extent, as shown in the case of 33-bus-S-O-Vm in Fig. 5. However, the use of the big-M method can lead to failure when numerous constraints couple large coefficients with near-zero decision variables. Such scaling — common in noisy (where most residuals are amplified) or large-scale cases — results in ill-conditioning, which stalls the solver during refinement and prevents completion. This explains the failure of `OI` in the 33-bus-S-N case and in larger systems.

Second, we have extended the discussion regarding the DC-coefficient-optimization by incorporating the reference "Optimizing parameters of the DC power flow," as detailed below:

Section IV.C.1. Coefficient Optimization

Parameters in a PPFL model can be further optimized in a data-driven manner to improve linearization accuracy [25], [26], [29]. In the DC power flow model, for instance, rather than relying solely on coefficients derived from physical line parameters, one can retain the DC model structure while treating the coefficients as trainable parameters. These coefficients can then be optimized using any of the DPFL training algorithms reviewed earlier. For instance, [29] formulates this tuning process as an unconstrained optimization problem, where a loss function measuring the discrepancy between the DC and AC power flow results is minimized (i.e., the same objective formulation principle as in Section III-C). For solutions, gradient-based algorithms, such as quasi-Newton methods implemented in MATLAB's fminunc solver, including BFGS and L-BFGS, can then be applied to optimize the coefficients. Similar ideas are also explored in [25], [26], where either the DC model or the state-independent PPFL model introduced in [30] is employed for data-driven parameter optimization.

Finally, we have incorporated the concept of localized training, as introduced in the two references: "Localized Structure in Secondary Distribution System Voltage Sensitivity Matrices" and "Distributed Measurement-Based Optimal DER Dispatch with Estimated Sensitivity Models," as detailed below:

Section VIII.3. Computational Efficiency

Over 20 methods are computationally intensive, especially for systems with more than 1000 buses. While accurate, many are unsuitable for real-time applications due to their reliance on optimization, nonlinear fitting, or iterative procedures. Improving optimization efficiency, accelerating convergence, enhancing nonlinear modeling, and managing large datasets remain key directions for enabling real-time DPFL. One promising approach is to leverage the grid's physical structure for localized computation. For example, [31] partitions the system into physically defined regions, estimates local models via standard regressions, and coordinates them through distributed optimization, thereby significantly reducing computation without sacrificing accuracy. More recently, [32] pushes localization further by working at the level of secondary load buses, exploiting the natural block structure of voltage sensitivity matrices. Each block is estimated independently using least-squares or ridge regression, achieving substantial speedups with minimal accuracy loss. Though these works do not introduce new DPFL training algorithms, they highlight how physically guided, modular modeling can serve as a scalable alternative to traditional, optimization-heavy approaches. While such methods rely on clear physical separability, they represent a promising direction toward real-time-capable DPFL solutions.

Comment 3.5: I recommend that the authors condense the review, back it with simulations and examples and submit it as a 1 part paper. My major suggestion is to combine the two papers into one, remove high-level explanations that are common knowledge in the domain in the theory part, condense the simulation section and remove the discussion part in the simulation section. I think the results should speak for themselves.

Response to Comment 3.5: As suggested by the reviewer, we have merged the previously separate theory and simulation papers into a single, unified manuscript. Also, thanks to the Editor-in-Chief's approval, we were granted an extended page limit of 16 pages. This support has allowed us to strike a balance between conciseness and comprehensiveness, enabling us to present a self-contained resource on data-driven power flow linearization methods — this has always been our goal, i.e., to ensure that readers can understand the key methodologies and their evaluation results, without needing to consult numerous external references.

Specifically, in response to the reviewer's advice, we have made significant revisions throughout the paper, including:

1) **Merging the two manuscripts** into a single document, using the simulation-part manuscript as the main foundation, while integrating key theoretical content.

2) **Condensing the theoretical content**, delivering concise yet precise descriptions of each method's formulation, assumptions, and design motivation.
3) **Condensing the simulation section**, with a greater emphasis on letting results speak for themselves, while maintaining enough discussion to aid interpretation.
4) **Shortening the discussion of future challenges** to reflect the most critical insights without redundancy.
5) **Rewriting the Abstract, Introduction, and Conclusion** to reflect the new unified structure and clarify the contributions of the revised manuscript.

We believe the revised manuscript presents a more focused and impactful contribution, while still serving as a comprehensive reference for data-driven power flow linearization. Given the extensive revisions made throughout the paper, we have not duplicated the updated content in this response letter. Instead, we kindly invite the reviewer to refer directly to the whole revised manuscript.

Comment 3.6: One of the motivating features of the authors for this work was the general lack of open-source codebases. Yet, even though the authors have made extensive comparisons, I could not find their open-source implementations. I recommend they open-source their work.

Response to Comment 3.6: We have open-sourced our work, and beyond that, we have further developed a comprehensive, user-friendly toolbox called DALINE based on this work, which integrates all the evaluated methods from the manuscript. The official website for DALINE is available here, covering the toolbox, a 150-page user manual, and all the other relevant documentation. With DALINE, all evaluations presented in this manuscript can be reproduced with just a few lines of code.

More specifically, DALINE includes 58 linearization methods, comprising 54 data-driven approaches and 4 widely-used physics-driven baselines. Beyond the methods themselves, DALINE also supports a full data-driven modeling pipeline with the following key functionalities: (1) Data Generation. (2) Data Pollution. (3) Data Cleaning. (4) Data Normalization. (5) Method Selection. (6) Method Customization. (7) Model Linearization. (8) Model Evaluation. (9) Result Visualization.

In fact, this manuscript and the toolbox are part of a broader research effort: the manuscript focuses on methodological analysis and practical evaluation, serving as a guidance resource for readers; DALINE, in contrast, introduces a user-friendly, practical software tool that allows researchers to rapidly apply data-driven power flow linearization methods, i.e., its primary role is to provide an application-oriented solution for researchers.

In the revised manuscript, we have included the link to DALINE to ensure accessibility for readers.

Section VII.B. Evaluation Overview

...

To facilitate reproducibility and further research, we have open-sourced all the evaluated methods presented in this work through a dedicated toolbox, DALINE, which is openly available at here, along with a 150-page user manual and all relevant documentation.

Comment 3.7: Further, I found no definitions for various metrics used to develop Fig. 2 to 4. Consider describing the metrics used to develop numbers in the results figures such that those numbers are reproducible.

Response to Comment 3.7: We agree that the clarity and reproducibility of evaluation metrics are critical. In fact, the definitions of the metrics used in Figs. 2–4 were already provided in the corresponding figure captions. For completeness and improved readability, we reproduce them below:

- **Fig. 2**: Computational times of evaluated methods under different test cases (unit: seconds; decimal values for times exceeding 10 seconds are disregarded).
- **Fig. 3**: Linearization accuracy ranking of all methods w.r.t. active branch flows, evaluated under the 118-bus-L-N case. The relative error of active branch flow i in sample t is computed by $\epsilon_{it} = |y_{it}^{true} - y_{it}^{method}|/|y_{it}^{true}|$. The mean relative error is thus the average of ϵ_{it} for $\forall i$ and $\forall t$, while the maximal relative error is the maximum of ϵ_{it} for $\forall i$ and $\forall t$. The methods are ranked based on their mean relative errors.
- **Fig. 4**: Rankings of the linearization accuracy of 45 methods w.r.t. active branch flows, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

Comment 3.8: The underlying assumption is that grid topology does not change when developing synthetic time-series measurements. However, over a period of 24 hours, general grid topologies change for real-world networks. How do authors recommend addressing this challenge?

Response to Comment 3.8: We agree that topological changes, such as switching actions or other remedial operations, frequently occur in real-world grids and can significantly affect model accuracy. However, we would like to clarify that this work is a review paper of existing data-driven power flow linearization methods, and we do not intend to propose new strategies to address the topological variation.

Yet, given the importance of this issue raised by the reviewer, we identify this as a critical open challenge for future research, Accordingly, we have extended the Challenges section to reflect it, as detailed below:

> **Section VIII.8. Topological Variation**
>
> Topology changes, such as line switching or phase-shifting, are common in power system operations, but they can invalidate models trained on pre-change data. Treating topology as a dynamic predictor rather than a fixed parameter may improve model adaptability. However, practical and reliable methods for this approach remain underexplored. Sparse transitional data and the growing frequency of switching, driven by renewable integration, further underscore the need for a robust, theory-backed framework to embed topology dynamics into DPFL models.

Comment 3.9: Finally, I recommend authors work on larger networks, at least those representative of major continental-sized networks. For smaller-sized networks, the value proposition for linear power flow models is significantly lower.

Response to Comment 3.9: In response, we have extended the test case pool to incorporate the 1888-bus system, which represents a snapshot of the VHV French transmission network provided by RTE. Additional evaluations have been conducted using this system under various settings, and the corresponding results and discussions have been added to the revised manuscript. Related revisions are detailed below:

Section VII. Numerical Evaluations

...

TABLE IV: Settings for the Test Cases

Test Case	Training Samples	Testing Samples	Fluctuation Level	Grid Type	Noise Level◇	Outlier Level◇
9-bus-S	150	100	95% - 105%	Transmission	-	-
9-bus-L	150	100	80% - 120%	Transmission	-	-
14-bus-S	200	150	95% - 105%	Transmission	-	-
14-bus-L	200	150	80% - 120%	Transmission	-	-
33-bus-S	300	200	95% - 105%	Distribution	-	-
33-bus-S-N	300	200	95% - 105%	Distribution	45dB	-
33-bus-S-O	300	200	95% - 105%	Distribution	-	2%
33-bus-L	300	200	80% - 120%	Distribution	-	-
39-bus-S	300	200	95% - 105%	Transmission	-	-
39-bus-L	300	200	80% - 120%	Transmission	-	-
69-bus-S	300	200	95% - 105%	Distribution	-	-
69-bus-L	300	200	80% - 120%	Distribution	-	-
118-bus-S	400	300	95% - 105%	Transmission	-	-
118-bus-L	400	300	80% - 120%	Transmission	-	-
118-bus-L-N	400	300	80% - 120%	Transmission	45dB	-
118-bus-L-O	400	300	80% - 120%	Transmission	-	2%
141-bus-S	400	200	95% - 105%	Distribution	-	-
141-bus-L	400	200	80% - 120%	Distribution	-	-
1354-bus-S	3000	1000	95% - 105%	Transmission	-	-
1354-bus-L	3000	1000	80% - 120%	Transmission	-	-
1888-bus-S	4000	1000	95% - 105%	Transmission	-	-
1888-bus-L	4000	1000	80% - 120%	Transmission	-	-

‡The terminal voltages of generators are fixed during data generation.
◇Noise and outliers are added only to the training data; testing data remains clean. Noise is modeled as white Gaussian at 45 dB [33], with each data point independently perturbed, simulating measurements from separate devices. For outliers, 2% of values in each variable (column) are randomly doubled, representing device-specific errors. Although this rate is small, the combined effect across variables increases the likelihood that a sample (row) contains at least one outlier.

...

Section VII.B. Evaluation Overview

...

... Fig. 6 highlights performance on the 1888-bus system...

...

Section VII.D.1. DPFL Performance vs. PPFL Performance

In the illustrative 118-bus example of Fig. 3, the gap between DPFL and PPFL reaches two orders of magnitude. Further, Figs 4–5 show that, in most cases, DPFL methods yield lower errors than the four PPFL baselines. Although these results already cover large systems such as the 1354-bus case, which represents part of the European transmission grid, Fig. 6 further evaluates an even larger system—the 1888-bus French transmission grid—where DPFL methods continue to significantly outperform PPFL approaches. Among the PPFL methods, only `TAY` occasionally approaches the accuracy of DPFL. These results indicate that DPFL methods can match—and often surpass—classical PPFL approaches, highlighting their potential and motivating further research in this direction.

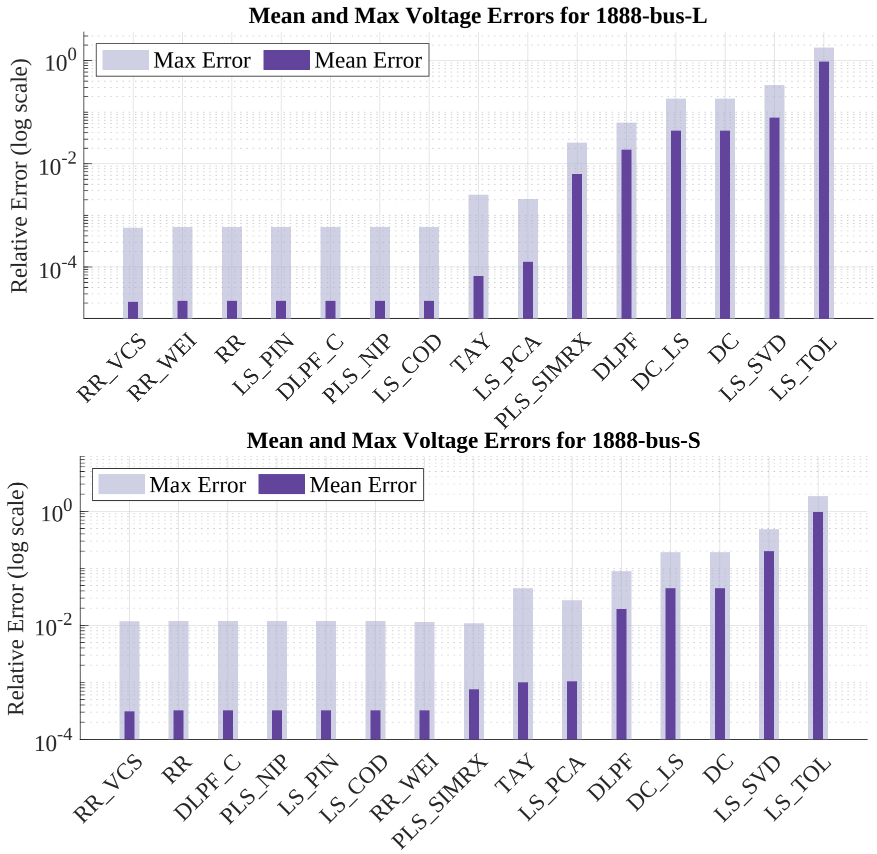

Fig. 6: Relative voltage errors of representative methods under the 1888-bus system. The upper panel corresponds to the system with larger fluctuation levels, while the lower panel represents the system with smaller fluctuations.

Comment 3.10: There are significant numbers of typos in the paper. Please run the paper through grammar and spelling check.

Response to Comment 3.10: We have thoroughly proofread the manuscript, and corrected all identified typographical and grammatical errors using grammar and spelling tools.

Comment 3.11: I could generally not follow predictor and response applicability in Section III.A. The application to the proposed work was not generally clear to me. Also what is ”arbitrary” in the Table II?

Response to Comment 3.11: The term “predictor and response applicability” refers to the flexibility of a DPFL method in selecting input (predictor) and output (response) variables during model training. Specifically, it evaluates whether a method can work with any measured variables as predictors and any unknown variables as responses, or if it enforces specific variable types (e.g., power injections only) due to algorithmic or structural constraints.

We also would like to clarify that the title of Section VI, “Applicability Evaluation”, refers to the assessment of how flexibly each method can be used under different data and modeling conditions. Accordingly, this section evaluates whether methods can handle practical challenges such as limited predictor/response availability, constant data (e.g., fixed generator voltages), and other situations commonly encountered in power system datasets. Correspondingly, Table III (formerly Table II in the previous manuscript) summarizes the evaluations in Section VI. However, **neither**

Section VI nor Table III intends to discuss real-world applications of linearization methods, such as unit commitment.

In addition, to avoid confusion regarding the term “arbitrary,” we have replaced it with “unconstrained” throughout the manuscript, including Table III. We believe this term more accurately conveys the intended meaning. For the updated version of Table III, please refer to **Response to Comment 3.2**.

Finally, we have added the following clarifying sentence to the revised manuscript:

Section VI.A. Predictor and Response Applicability

DPFL approaches have the potential to accommodate unconstrained known variables as predictors, and unconstrained unknown variables as responses, meaning that the method imposes no built-in restriction on which measured variables may serve as predictors or responses during model training...

Response to Reviewer 4

General Comments to the Author: This two-part paper addresses the challenges of selecting the right data-driven power flow linearization (DPFL) method. The main contribution is the development of a unified framework which allows for systematic analysis and offers open-source comparisons. DPFL have growing popularity for their accuracy and adaptability, and as such, this benchmark seems highly interesting and relevant. The reviewer appreciates the depth of the work and the large number of different DPLF methods included in the study. Nevertheless, the reviewer has several comments/suggestions in order to improve the paper.

Response to the General Comment: We would like to thank the reviewer for the thoughtful and encouraging feedback. The reviewer's comprehensive observations and suggestions have led to meaningful revisions, both in content and structure, greatly improving the overall quality of our manuscript. In response, we have significantly revised and enhanced the manuscript to address all concerns. Major revisions include **(1) adding new evaluations** on noise robustness, including experimental results and discussions on real-world sensor data; **(2) incorporating and analyzing recent studies** on localized training, with emphasis on computational efficiency; **(3) clarifying the performance trade-offs** between physics-informed and purely data-driven models, supported by improved theoretical-experimental alignment; **(4) revising noise scenarios** to reflect realistic conditions, with updated test settings and results; **(5) ensuring transparency in hyperparameter tuning**, with added documentation and remarks on conditional performance; **(6) expanding the discussion on limited observability**, outlining potential solutions and future research directions. For detailed responses and specific manuscript modifications, please refer to:

- **Response to Comment 4.1** for the reference to similar benchmark studies on physics-driven linearization methods.
- **Response to Comment 4.2** for discussions on discarding outdated data and its potential for computational efficiency.
- **Response to Comment 4.3** for clarifications on the validity and scope of Models 1, 2, and 3.
- **Response to Comment 4.4** for the added evaluations on noise robustness and discussions on the noise level of real-world sensor data.
- **Response to Comment 4.5** for concrete examples of mapping functions in Model 3.
- **Response to Comment 4.6** for discussions on localized training and integration of referenced studies.
- **Response to Comment 4.7** for elaborations on the performance of physics-informed versus purely data-driven models.
- **Response to Comment 4.8** for corrections of noted typos and textual refinements.
- **Response to Comment 4.9** for clarifications and revisions of noise scenarios to reflect realistic settings.
- **Response to Comment 4.10** for clarifications on the scaling factor methodology and its relevance.
- **Response to Comment 4.11** for transparency regarding hyperparameter tuning and relevant modifications.
- **Response to Comment 4.12** for the expansion on the potential solution for the limited observability issue.
- **Response to Comment 4.13** for comprehensive typographical and grammar corrections.

The detailed, point-by-point responses are provided in the sections below.

Comment 4.1: Similar work: Are there any similar benchmarks for other applications in power systems which were developed? For example, the comparison of physics based power flow models? This should be integrated in the introduction.

Response to Comment 4.1: To address this point, we have integrated a reference to one relevant and comprehensive benchmark study in the field: a systematic tutorial on physics-driven power flow linearization methods. While this work was cited in the previous version, we have now explicitly highlighted it at the beginning of the Introduction to provide clearer context.

Section I. Introduction

Linear power flow models are fundamental to power systems, widely applied in both academic research and industrial practice [34], [35], including in state estimation, unit commitment, economic dispatch, grid planning, and system control [25], underpinning trillion-dollar markets worldwide. Accordingly, power flow linearization methods, particularly physics-driven approaches, have been studied for decades; see [35] for a comprehensive monograph of these methods.

...

Comment 4.2: Limitation 4: The authors state that the need of continuous updates with new observations is necessary, which incurs significant computational burden. At what point can old data be discarded? The problem does not necessarily need to increase in size if old observations are discarded.

Response to Comment 4.2: Indeed, discarding outdated data is a valuable idea that warrants further attention. Some methods partially address this issue. For example, the weighted recursive partial least squares method, discussed in Section II.B.2, employs a forgetting factor less than 1, recursively applied to past data points. This gradually reduces the influence of older data without explicitly discarding them.

However, it remains an open question when, and under what conditions, old data can be fully discarded in a principled and reliable way. Currently, no established unlearning framework exists for DPFL methods that specifies clear thresholds or criteria for safely removing past observations without affecting model accuracy or stability.

We agree that developing such criteria would be highly valuable, particularly for real-time or resource-constrained applications. Accordingly, we have included this as a future research direction in the revised manuscript, under the discussion on improving computational efficiency.

Section VIII.3. Computational Efficiency

...

Another promising direction is to develop principled data management strategies for discarding outdated observations. While recursive methods reduce the influence of past data via forgetting factors, there is no established framework in DPFL for explicitly removing old data without affecting model performance. Defining such criteria could greatly enhance computational efficiency in real-time and resource-constrained applications.

Comment 4.3: In line with Limitation 4: is the goal to train a global linear power flow model which is valid across all operating points, or a small signal model which is only valid around the operating point. Maybe the authors can specify this when Model 1, 2 and 3 are introduced.

Limitation 5: Is this not inherently the case for all linear power flow models? Are Model 2 and 3 potentially superior in this regard?

Response to Comment 4.3: Generally, Model 1, as a single linear model, is typically valid near the training operating point, capturing small-signal behavior. Models 2 and 3, however, offer greater flexibility: Model 2 extends validity across broader operating regions through segmentation, while Model 3 approximates nonlinear behavior via feature mapping. Although all these models inherently simplify the nonlinearity of AC power flows, Models 2 and 3 may better capture nonlinear effects, depending on how segmentation or mapping is applied. This is consistent with the experimental results in Figs. 4–5, where methods such as `RR_KPC` and `PLS_CLS` (Model 2), as well as `LS_LIFX` and `LS_LIFXi` (Model 3), frequently achieve top performance across diverse test cases.

In the revised manuscript, as recommended by the reviewer, we have added further clarification on the validity range and nonlinearity-handling capabilities of Models 1, 2, and 3, at the place where they are introduced:

Section I. Introduction $\rightarrow$ Notation

Model 1 is a single linear model with $\boldsymbol{\beta} \in \mathbb{R}^{N_x \times N_y}$ being the coefficient matrix (we denote $\boldsymbol{\beta}_j \in \mathbb{R}^{N_x \times 1}$ as the j-th column of $\boldsymbol{\beta}$ hereafter). This model is typically valid as a small-signal, local model around the operating point used for training. **Model 2** is a piecewise linear model, where $\boldsymbol{\beta}(k) \in \mathbb{R}^{N_x \times N_y}$ is the coefficient matrix for segment k, and K denotes the number of total segments. This model, by combining local models, can potentially extend validity across a wider range of operating conditions, and capture non-linearities more effectively than a single linear model. **Model 3**, parameterized by $\boldsymbol{\beta}_\phi \in \mathbb{R}^{N_\phi \times N_y}$, describes the linear relationship between $\boldsymbol{y}$ and $\phi(\boldsymbol{x})$, where $\phi \colon \mathbb{R}^{N_x} \rightarrow \mathbb{R}^{N_\phi}$ is a mapping function, such as a kernel transformation, dimension-lifting, or a simple voltage square $\phi(\boldsymbol{v}) = \boldsymbol{v}^2$. Depending on the choice of mapping and data coverage, this model may also approximate nonlinear behaviors and broaden applicability beyond local operating regions.

Comment 4.4: Noise level: Can the authors quantify the noise level with currently available sensors? What are the minimum requirements for sensors in order to feed data-driven power flow models?

Response to Comment 4.4: In response to this, we have added an additional experiment on noise robustness; see Section VII.D.2.C (Data Pollution). Specifically, we evaluated the performance of representative DPFL methods under varying signal-to-noise ratio (SNR) levels, ranging from 60 dB to 30 dB. As illustrated in Fig. 8, all methods achieve average voltage errors below 2×10^{-3} at an SNR of 45 dB.

To base this benchmark in real-world conditions, we refer to the empirical study by Brown et al. [33], which characterizes measurement noise in PMU data across high-, medium-, and low-voltage systems. Their results show that the SNR values for voltage measurements consistently fall within the 44–47 dB range, and they recommend using 45 dB as a representative value for PMU-grade data. Based on this, we adopt 45 dB as a realistic and practically relevant benchmark in our analysis.

These findings suggest that data-driven power flow models can be effectively trained using measurements from currently available PMUs. Accordingly, sensors capable of delivering an SNR around 45 dB, or equivalently, appear sufficient to support accurate and dependable DPFL model development in practice.

Detailed revisions made to the manuscript are given below:

Section VII.D.2.C. Data Pollution

...

Although noise- or outlier-robust methods (e.g., `OI`, `LS_HBLD`, and `LS_HBLE`) help mitigate the impact of data pollution, all DPFL approaches exhibit noticeable error growth—up to three orders of magnitude—under increasing noise or outlier levels. These observations underscore the continued importance of data pre-processing. Nevertheless, from a practical standpoint, DPFL methods can still yield satisfactory linear models. As shown in Fig. 8, while the average voltage error increases as the signal-to-noise ratio (SNR) decreases from 60 dB to 30 dB, all tested methods reduce their errors to around 2×10^{-3} at 45 dB. This SNR level is supported by practical studies of field PMU data across different voltage levels, where measured SNRs consistently fall within the 44–47 dB range [33]. These findings suggest that DPFL methods are compatible with realistic measurement conditions and hold strong potential for practical deployment.

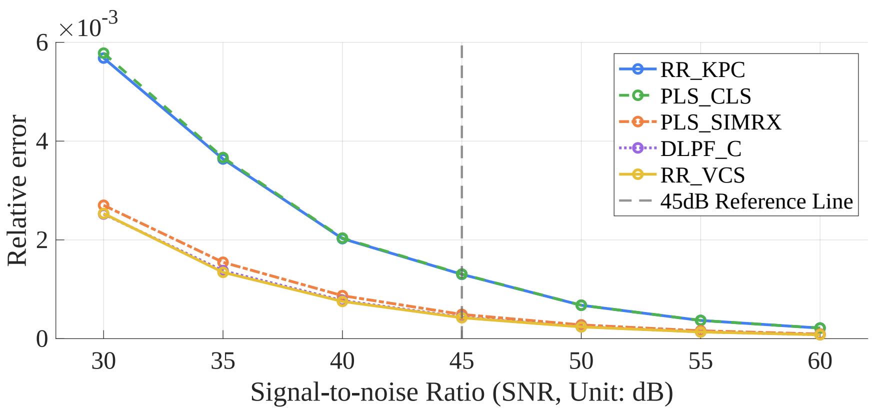

Fig. 8: Average voltage relative errors (in average) of representative methods on the 118-bus-L system under varying signal-to-noise ratios.

Comment 4.5: In the introduction when the mapping function $\phi(\boldsymbol{x}$ is introduced, it would be useful to provide a few examples what this mapping function might do (i.e. square voltage levels, etc.)

Response to Comment 4.5: In response, we have revised the description of Model 3 in the manuscript to include several concrete examples of $\phi(\boldsymbol{x})$:

Section I. Introduction → Notation

Model 3, parameterized by $\boldsymbol{\beta}_\phi \in \mathbb{R}^{N_\phi \times N_y}$, describes the linear relationship between $\boldsymbol{y}$ and $\phi(\boldsymbol{x})$, where $\phi\colon \mathbb{R}^{N_x} \to \mathbb{R}^{N_\phi}$ is a mapping function, such as a kernel transformation, dimension-lifting, or a simple voltage square $\phi(\boldsymbol{v}) = \boldsymbol{v}^2$. Depending on the choice of mapping and data coverage, this model may also approximate nonlinear behaviors and broaden applicability beyond local operating regions.

Comment 4.6: The authors mention that linear power flow models follow a structure of the Jacobian matrix. In fact, physics informed power flow models likely end up with the structure of the Jacobian. Particularly for large systems which are radial or weakly meshed, the Jacobian has a high sparsity. In such a setting a DPFL training algorithm is challenged by the large dimension of the problem, and the fact that many of the elements are actually zero. Several works have explored the training of "local" linear power flow models which can be combined to form the full linear power flow model. These methods do not train the full model in one batch. This may bring advantages in computation and accuracy. The authors should explore this topic and potentially include it in the study. The following references and others should be considered:

1) Samuel Talkington, Santiago Grijalva, Matthew J. Reno, Joseph A. Azzolini, Jouni Peppanen, "Localized structure in secondary distribution system voltage sensitivity matrices, Electric Power Systems Research," Volume 226, 2024, 109788, ISSN 0378-7796, https://doi.org/10.1016/j.epsr.2023.109788.
2) S. Nowak, Y. C. Chen and L. Wang, "Distributed Measurement-Based Optimal DER Dispatch With Estimated Sensitivity Models," in IEEE Transactions on Smart Grid, vol. 13, no. 3, pp. 2197-2208, May 2022, doi: 10.1109/TSG.2021.3139450.

***Response to Comment 4.6**:* We fully agree on the importance of localized training strategies, particularly for large-scale, sparse systems where the Jacobian structure can be effectively exploited. The localized training approach highlighted by the reviewer offers significant potential for enhancing computational efficiency and scalability in data-driven power flow linearization.

Regarding the two referenced works, Talkington et al. [32] and Nowak et al. [31] provide valuable insights into leveraging physical grid structures to partition the training process. Specifically:

- Talkington et al. [32] employ ridge regression with regularization — as discussed in Section II.C of our manuscript — to independently estimate local voltage sensitivity matrices at the level of secondary load buses, exploiting block sparsity for improved efficiency.
- Nowak et al. [31] utilize weighted recursive partial least squares — as discussed in Section II.B of the manuscript — to train local models within physically defined regions, coordinating them through distributed optimization.

While these studies do not intend to propose new linearization training algorithms per se, their structural exploitation of the grid enables localized training, which avoids the computational burden of full-system batch learning. To recognize the value of these contributions, we have incorporated a detailed discussion of these works into the future research directions section, specifically under the topic of improving computational efficiency:

Section VIII.3. Computational Efficiency

...

One promising direction is to leverage the grid's physical structure for localized computation. For example, [31] partitions the system into physically defined regions, estimates local models via standard regressions, and coordinates them through distributed optimization, thereby significantly reducing computation without sacrificing accuracy. More recently, [32] pushes localization further by working at the level of secondary load buses, exploiting the natural block structure of voltage sensitivity matrices. Each block is estimated independently using least-squares or ridge regression, achieving substantial speedups with minimal accuracy loss. Though these works do not introduce new DPFL training algorithms, they highlight how physically guided, modular modeling can serve as a scalable alternative to traditional, optimization-heavy approaches. While such methods rely on clear physical separability, they represent a promising direction toward real-time-capable DPFL solutions.

...

Comment 4.7: The authors mention that the physics informed models may perform better. How many physical parameters are required or desired for an improvement? At the same time, the authors argue, that the data-driven models outperform purely physics informed models (in the simulation part). This seems somewhat contradicting. Can you elaborate on this?

Response to Comment 4.7: We appreciate this insightful comment and would like to clarify our intent more explicitly.

First, our statement refers to comparisons *within* a given method, i.e., a data-driven method with physics-informed features may perform better than its purely data-driven counterpart, but not necessarily better than all other methods. Specifically:

- **Physics-enhanced vs. pure version of the same method:** A physics-informed variant does not always outperform its purely data-driven version. For instance, `LCP_COU` integrates accurate physical information (e.g., nodal angle signs) and achieves better performance than `LCP_COUN`, as discussed in Section VII.D.3.G. Conversely, `LCP_BOX` uses an approximated physical model and performs worse than its pure data-driven version `LCP_BOXN` (Section VII.D.3.H), illustrating that inaccurate physics can degrade results.
- **Physics-informed vs. general data-driven methods:** Overall, physics-informed methods do not outperform purely data-driven methods. As shown in Figs. 4–5, general data-driven approaches often yield superior accuracy. This is due to potential drawbacks of embedding physical constraints: (i) excluding informative variables, (ii) conflicts with data normalization, (iii) reliance on potentially inaccurate physical inputs, and (iv) reduced flexibility in model structure (see Section VIII.5).

Regarding the number of physical parameters required for improvement, it depends on the specific method and system; e.g., `LCP_COU` benefits from knowledge of grid topologies, the complexity of which varies by grids.

Finally, to avoid possible contradictions and to strengthen the alignment between theoretical and experimental content, we have made the following revisions:

1) We have merged the theory and simulation papers into a single, coherent manuscript. As a result, the **Abstract**, **Introduction**, and **Conclusion** have been updated for better alignment.
2) A comprehensive table summarizing the capabilities and limitations of all reviewed methods has been added (**Response to Comment 2.1**), directly linking theoretical claims to numerical observations.

3) We have improved the correspondence between theory and results throughout the manuscript, notably between the summary table, **Section VI** (Applicability Evaluations), and **Section VII** (Numerical Evaluations).

Due to the extensive nature of these revisions, we have not duplicated all changes here, except for the summary table shown below. We kindly invite the reviewer to consult the revised manuscript for a full view.

TABLE I: Capabilities and Limitations of DPFL Approaches

References	Methodology	Capabilities	Limitations
[2]	Ordinary Least Squares	Easy to implement with a clear, interpretable solution.	L1: affected by multicollinearity; L2: outlier-sensitive; L3: assumes heteroskedasticity; L4: costly updates; L5: ignores non-linearity; L6: noise-sensitive.
[3], [4]	Least Squares with COD/SVD	Handles multicollinearity; faster due to reduced triangular factors.	Zero singular values may still hinder inversion; L2 to L6.
[5], [6]	Least Squares with Huber Loss	More tolerant to outliers; alleviates multicollinearity impact.	No closed-form; computationally heavy with hyperparameters tuning; no guarantee on noise efficacy; L3; L4; L5.
[7]	Generalized Least Squares	Accounts for heteroskedasticity & autocorrelation when covariance of residuals known.	True covariance hard to obtain; needs strong assumptions; L1; L2; L4 to L6.
[8]	Total Least Squares	Consider predictors' observational noise as well.	Assumes equal noise variance from various devices; rank-deficiency unresolved; L3 to L5.
[8]	Weighted Total Least Squares	Allows unequal noise treatment for the noise of both predictors and responses.	Non-convex; heuristic approximations; computationally heavy; may reach local optima; L3 to L5.
[9]	Recursive Least Squares	Fast incremental updates; adapts to non-stationarity.	L1 to L3; L6.
[10]	Ordinary Partial Least Squares	Designed to tackle multicollinearity.	Needs latent-dimension tuning; L2 to L6.
[11]	Weighted Recursive Partial Least Squares	Efficient online update; handles multicollinearity.	L2; L3; L5; L6
[12]	Ordinary Ridge Regression	Simple; handles multicollinearity; interpretable.	Penalty tuning expensive; L2 to L6.
[13]	Locally Weighted Ridge Regression	Alleviates non-linearity and multicollinearity with simple closed form.	Requires tuning of more hyperparameters; L2 to L4; L6.
[14]	Clustering-based Ridge Regression	Improves fit to nonlinear patterns; handles multicollinearity.	Many hyperparameters; possible bias and underdetermined clusters; complexify integration into applications; L2 to L4; L6.
[15]	Ordinary Support Vector Regression	Naturally regularized; tolerant to minor outliers; alleviates multicollinearity impact.	Computationally heavy for large systems; hyperparameters tuning expensive; L3 to L5.
[16], [17]	Kernel Support Vector Regression	Alleviates non-linearity while retaining the support vector regression's strengths.	Kernels may not always create a more linear space; high computation with hyperparameters tuning; L3; L4.
[8], [18]	Linearly Constrained Programming (Bound constrained; Structure constrained; Coupling constrained)	Remains convex; leverages physical insight; reduces multicollinearity impact.	May omit valuable predictors; sensitive to scaling; accuracy depends on physical fidelity; limited response set; L2; L4 to L6.
[19]	Chance-constrained Programming	Reduces hyperparameters; clear statistical meaning; robust to small outliers; less multicollinearity risk.	Allows large errors; feasibility sensitive to confidence; big-M method increases problem's scale; computationally heavy for large systems; L3 to L5.
[20]	Distributionally Robust Chance-constrained Programming	Explicit worst-case error control; general robust framework; multicollinearity tolerant.	Needs a preset operating point; computationally heavy for large systems; L3 to L5.
[21]	Mixed-integer Programming	Explicit outlier-budget with clear statistical meaning; robust as long as only a few rows are corrupted; solvable with any linear programming/barrier solver.	Relies on a Big-M constant that can cause severe scaling; many tiny fractional columns explode the condition number, so crossover may stall; memory- and time-demanding for large networks; L3 to L5.
[22]	Variable Bundle Strategy	Handles changing bus types dynamically; increases model flexibility.	May cause large linearization errors if key matrices become near-singular.
[2], [12], [14], [15], [18], [19]	Voltage Squaring	Easy to implement; commonly improves model linearity; widely recognized.	No major limitations.
[12]	Voltage-Angle Coupling	Produces interpretable linear models; introduces accurate physics.	May greatly increase the variable number with an unknown topology.
[23]	Dimension Lifting	Flexible transformation; alleviates non-linearity.	Hard to apply in application tasks like dispatch; it increases model complexity.
[10], [24]–[26]	Physical Model Integration (Coefficient Optimization; Error Correction)	Improves accuracy when high-fidelity physical models are available; improves data efficiency.	Sensitive to incorrect physical assumptions; optimization-based methods are computationally heavy.

◇The unshaded rows represent training algorithms, while the shaded ones indicate supportive techniques.
†For illustrative numerical examples, please see Section VII-C, where representative failure cases are identified and analyzed in detail.
‡Table III provides a structured applicability assessment across multiple dimensions, aiding the identification of potential failure scenarios.
⋆Although noise may statistically promote full-rankness, L1 and L6 still represent distinct and practically relevant limitations.

Comment 4.8: Minor comments/typos: Page 1/line 54: including active and reactive power injections of PQ buses? Page 3/line 28: "some these."

Response to Comment 4.8: In the revised manuscript, we have addressed both comments as follows:

- Page 1, Line 54: The sentence has been revised to correctly state the elements in $\widetilde{\boldsymbol{x}}$:

  Section I. Introduction → Notation

  ...

  We define $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{\widetilde{N}_x}$ as the vector of $\widetilde{N}_x$ independent variables, including active power injections at PQ and PV nodes, reactive power injections at PQ nodes, voltage magnitudes at PV nodes, and the voltage phase angle of the slack node.

  ...

- Page 3, Line 28: The sentence containing the typo "some these" has been removed entirely due to the restructuring of that section, which now correctly incorporates the summary table discussed in **Response to Comment 4.7**.

Comment 4.9: Experiment settings: The "joint noise" case (this implies that the entire system's data were measured by a single device). This sounds merely practical. What were the intentions here? Does this mean that each node measurement is subject to the exact same measurement error? Would it not make more sense that nodes are subject to the same error distribution?

Response to Comment 4.9: We agree that the terminology and intent were not sufficiently clear, and that the setup could be seen as impractical.

Our original intention was to investigate the relative impact of noise and outliers applied uniformly across all variables (i.e., "joint") versus applied independently per variable (i.e., "individual"). The results aimed to emphasize how different noise modeling assumptions can affect DPFL method performance, thereby encouraging greater transparency in how noise is introduced in benchmarking studies.

However, we now recognize that this setup may cause more confusion than insight. Therefore, in the revised manuscript, we have removed all simulation results involving "joint noise" and "joint outliers." Instead, we focus exclusively on the more realistic and practically relevant "individual noise" and "individual outlier" scenarios. These reflect standard conditions, where each variable is measured independently by separate devices, each subject to the same underlying error distribution.

The revised test case settings and updated simulation results now exclude the "joint" scenarios entirely, as detailed below:

Section VII.A. Experiment Settings

...

TABLE IV: Settings for the Test Cases

Test Case	Training Samples	Testing Samples	Fluctuation Level	Grid Type	Noise Level◇	Outlier Level◇
9-bus-S	150	100	95% - 105%	Transmission	-	-
9-bus-L	150	100	80% - 120%	Transmission	-	-
14-bus-S	200	150	95% - 105%	Transmission	-	-
14-bus-L	200	150	80% - 120%	Transmission	-	-
33-bus-S	300	200	95% - 105%	Distribution	-	-
33-bus-S-N	300	200	95% - 105%	Distribution	45dB	-
33-bus-S-O	300	200	95% - 105%	Distribution	-	2%
33-bus-L	300	200	80% - 120%	Distribution	-	-
39-bus-S	300	200	95% - 105%	Transmission	-	-
39-bus-L	300	200	80% - 120%	Transmission	-	-
69-bus-S	300	200	95% - 105%	Distribution	-	-
69-bus-L	300	200	80% - 120%	Distribution	-	-
118-bus-S	400	300	95% - 105%	Transmission	-	-
118-bus-L	400	300	80% - 120%	Transmission	-	-
118-bus-L-N	400	300	80% - 120%	Transmission	45dB	-
118-bus-L-O	400	300	80% - 120%	Transmission	-	2%
141-bus-S	400	200	95% - 105%	Distribution	-	-
141-bus-L	400	200	80% - 120%	Distribution	-	-
1354-bus-S	3000	1000	95% - 105%	Transmission	-	-
1354-bus-L	3000	1000	80% - 120%	Transmission	-	-
1888-bus-S	4000	1000	95% - 105%	Transmission	-	-
1888-bus-L	4000	1000	80% - 120%	Transmission	-	-

‡The terminal voltages of generators are fixed during data generation.

◇Noise and outliers are added only to the training data; testing data remains clean. Noise is modeled as white Gaussian at 45 dB [33], with each data point independently perturbed, simulating measurements from separate devices. For outliers, 2% of values in each variable (column) are randomly doubled, representing device-specific errors. Although this rate is small, the combined effect across variables increases the likelihood that a sample (row) contains at least one outlier.

...

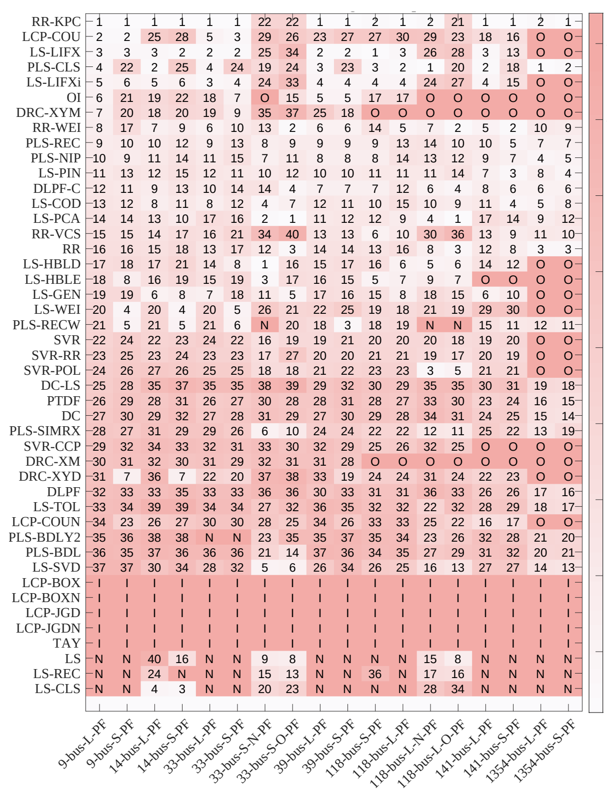

Fig. 4. Rankings of the linearization accuracy of 45 methods w.r.t. active branch flows, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

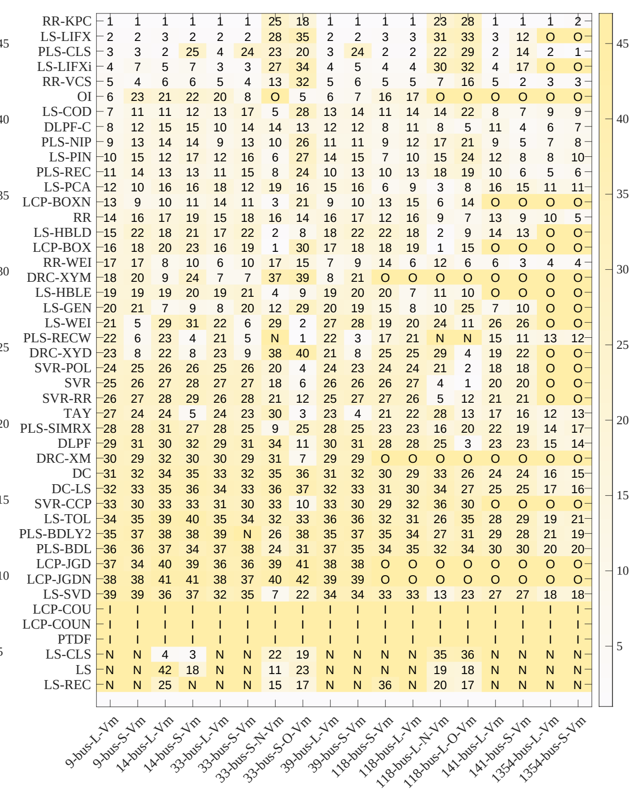

Fig. 5. Rankings of the linearization accuracy of 45 methods w.r.t. voltage magnitudes, with "1" being the most accurate. "I", "N", and "O" refer to the INA failure, the NaN failure, and the OOT failure, respectively.

Comment 4.10: Scaling factor: Are all nodes subject to the same scaling factor, or have different nodes different factors in one scenario? Different scaling factors per nodes would seem more practical.

Response to Comment 4.10: We fully agree that applying different variation factors per node is a practical way to reflect localized fluctuations. Indeed, several studies [3], [18], [20] adopt this approach by independently adjusting power injections at each node using distinct random factors.

However, using a uniform scaling factor across all nodes within a sample is also a well-recognized practice, particularly in system-level studies, e.g., in [2], [14]. This is especially common when time-series data are employed as the basis for nodal power variation [2]. In our study, we follow the methodology of [2]: a normalized time-series curve provides a sequence of global scaling factors across time steps, each further perturbed by a distinct random fluctuation to capture variability. Each resulting time-series factor is then applied uniformly to the nominal active and reactive powers of all nodes within each time step.

To clarify this point, we have explicitly updated the caption of Fig. 1 to describe our scaling approach, as shown below:

Section VII.A. Experiment Settings

...

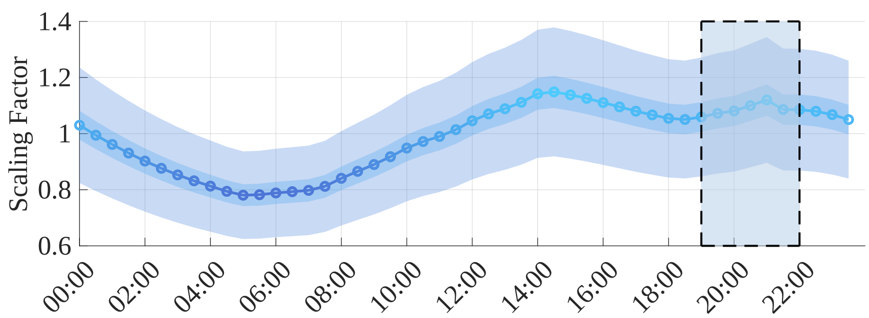

Fig. 1: The scaling factor curve reflects load and generation variability, focusing on the 19:00 to 22:00 interval. The time window is divided into N_s steps, where load and generation at each step are computed by multiplying the base value by the scaling factor and a random multiplier within the specified fluctuation range, consistent with the approach in [2].

...

Comment 4.11: Results: The authors mention that a change in the forgetting factor of the PLS_RECW has a significant result on the performance, or even the solvability of the problem. This also largely affects the results of a method. For example, there is very minimal difference between PLS_RECW and PLS_REC, however their rankings are significantly different. The reviewer has some concerns about the tuning of the different models for specific scenarios. To what extent the hyperparameters were tuned for a specific scenario? Would proper tuning provide significantly different results? It is suggested that the authors verify and optimize the hyperparameters, in order to make meaningful statements about the methods.

Response to Comment 4.11: We completely agree that hyperparameter sensitivity, especially for methods like `PLS_RECW`, is indeed an important consideration. **In our study, we tuned hyperparameters via cross-validation**

whenever the original references did not specify exact values. However, for methods where specific values were firmly stated in the literature, we adhered to those settings for consistency with prior work. For example, [11] explicitly recommends using 0.6 for the forgetting factor in `PLS_RECW`, which we followed accordingly.

We also emphasize that our study does not aim to establish absolute or definitive rankings among the evaluated methods. Rather, the reported results illustrate performance under particular testing conditions and chosen hyperparameters. As explicitly stated in the publicly accessible arXiv version of this manuscript [**?**], we acknowledge the conditional nature of these outcomes (although initially omitted due to page limits, this remark has been reinstated in the revised manuscript):

> *Remark: ... Additionally, it is important to note that no method is without flaws. The analysis of limitations presented in this paper is not intended as criticism but rather forms part of a comprehensive evaluation under certain scenarios with specified hyperparameters.*

Therefore, our intention is not to draw definitive conclusions, but rather to offer practical insights into performance trends and highlight potential challenges commonly encountered in practice. These insights serve to guide readers toward selecting methods most suitable for their specific applications.

To further support readers in directly applying and adapting these methods, we have developed an open-source toolbox, DALINE, integrating all evaluated approaches. This toolbox is accompanied by an extensive 150-page user manual and supplementary documentation, accessible via the official website. With DALINE, all evaluations described in this manuscript can be easily reproduced and customized, allowing exploration of diverse hyperparameter settings based on the identified performance trends from this manuscript.

Finally, we have provided a comprehensive summary table listing all hyperparameters employed in this study, along with their respective tuning ranges, where applicable. Detailed hyperparameter information is also conveniently accessible through DALINE. Due to strict page limits (initially restricted to 10 pages and currently 16 pages), this table is included only in the supplementary material [28] but is sufficiently cited in the revised manuscript. Below is the newly reinstated remark in the manuscript, and the hyperparameter table in the supplementary material:

> **Section I. Introduction → Remark**
>
> (i) No method is flawless. The following limitation analysis is not a criticism, but merely a part of a comprehensive assessment under certain scenarios with specified hyperparameters...
>
> **Supplementary Material → Experiment Settings**
>
> The reader is referred to the following table for a thorough illustration of the settings for each method, including their predictors, responses, hyperparameters, cross-validation ranges (in this study, cross-validation is employed for hyperparameter tuning only in cases where the literature does not specify the hyperparameter values), datasets, solvers (when applicable), and additional relevant information.

Approach	Predictor[◇]	Response[◇]	Miscellaneous	Solver
LS	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized[¶]	-
LS_SVD	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized	-
LS_COD	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized	-
LS_HBLD	P, Q, V^2, θ_{ref}	V, PF	$\delta^{HUB} \in \{0.01, 0.02, \cdots, 0.05\}$; $N_{cv} = 5$[†]; Data: Normalized	FMINUNC
LS_HBLE	P, Q, V^2, θ_{ref}	V, PF	$\delta^{HUB} \in \{0.01, 0.02, \cdots, 0.05\}$; $N_{cv} = 5$; Data: Normalized	MOSEK
LS_TOL	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized	-
LS_CLS	P, Q, V^2, θ_{ref}	V, PF	$k \in \{2, 3, \cdots, 10\}$; $N_{cv} = 5$; Data: Normalized	-
LS_LIFX	P, Q, V^2, θ_{ref}	V, PF	f_{lift}: Gauss; Data: Normalized	-
LS_LIFXi	P, Q, V^2, θ_{ref}	V, PF	f_{lift}: Gauss; Data: Normalized	-
LS_WEI	P, Q, V^2, θ_{ref}	V, PF	ϖ: 0.6; Data: Normalized	MOSEK
LS_REC	P, Q, V^2, θ_{ref}	V, PF	New Data Share = 40%[‡]; $\kappa = 0.99$; Data: Normalized	-
PLS_SIMRX	P, Q, V^2, θ_{ref}	V, PF	$N_p = \text{Rank(X)}$; Data: Normalized	-
PLS_BDL	P, Q, V^2	V, PF	$N_p = \text{Rank(X)}$; Data: Normalized	-
PLS_BDLY2	P, Q, V^2	V, PF	$N_p = \text{Rank(X)}$; Data: Normalized	-
PLS_REC	P, Q, V^2, θ_{ref}	V, PF	New Data Share = 40%; Data: Normalized	-
PLS_RECW	P, Q, V^2, θ_{ref}	V, PF	New Data Share = 40%; $\varpi = 0.6$; Data: Normalized	-
RR	P, Q, V^2, θ_{ref}	V, PF	$\lambda = 10^{-10}$; Data: Normalized	-
RR_VCS	P, Q, V^2	V^2, R_{ij}, C_{ij}	$\lambda = 10^{-10}$; Data: Original[¶]	-
RR_KPC	P, Q, V^2, θ_{ref}	V, PF	$\lambda = 10^{-10}$; $k \in \{2, 3, \cdots, 10\}$; $\eta \in \{10^2, 10^3, \cdots, 10^5\}$; $N_{cv} = 5$; Data: Normalized	-
RR_WEI	P, Q, V^2, θ_{ref}	V, PF	$\lambda = 10^{-10}$; $\tau \in \{0.1, 0.11, \cdots, 0.35\}$; $N_{cv} = 5$; Data: Normalized	-
SVR	P, Q, V^2, θ_{ref}	V, PF	$\epsilon = 10^{-4}$; $\omega = 10$; Data: Normalized	GUROBI
SVR_CCP	P, Q, V^2, θ_{ref}	V, PF	$\epsilon = 10^{-4}$; $\omega = 10$; $M = 10^6$; $\zeta_j^{CCP} = 95\%$; Data: Normalized	GUROBI
SVR_POL	P, Q, V^2, θ_{ref}	V, PF	$\epsilon = 10^{-4}$; Kernel Model: 3rd-order Polynomial; Data: Normalized	FITRSVM
SVR_RR	P, Q, V^2, θ_{ref}	V, PF	$\epsilon = 10^{-4}$; $\omega = 10$; $\lambda = 10^{-4}$; Data: Normalized	GUROBI
LCP_BOX	P, Q	V, θ	Bound: min/max coefficients of the 1st-order Taylor Approximation Model[§]; Data: Original	MOSEK
LCP_BOXN	P, Q	V, θ	Data: Original	MOSEK
LCP_COU	V, θ	PF	$\delta^{LIN} = 10^{-2}$; Data: Original	MOSEK
LCP_COUN	V, θ	PF	Data: Original	MOSEK
LCP_JGD	P, Q	V, θ	Structure: 1st-order Taylor Approximation Model; Data: Original	SDPT3
LCP_JGDN	P, Q	V, θ	Data: Original	SDPT3
DRC_XM	P, Q, V^2, θ_{ref}	V, PF	$\epsilon_j = 10^{-4}$; $\zeta_j^{DRC} = 95\%$; Data: Normalized	MOSEK
DRC_XYM	P, Q, V^2, θ_{ref}	V, PF	$\epsilon_j = 10^{-4}$; $\zeta_j^{DRC} = 95\%$; Data: Normalized	MOSEK
DRC_XYD	P, Q, V^2, θ_{ref}	V, PF	$\epsilon_j = 10^{-4}$; $\zeta_j^{DRC} = 95\%$; Data: Normalized	GUROBI
OI	P, Q, V^2, θ_{ref}	V, PF	$M = 10^6$; $\theta^{OI} = 0.9$; $p = 8\%$; Data: Normalized	GUROBI
DC_LS	P	θ	Data: Original	-
DLPF_C	P, Q, V, θ_{ref}	V, PF	Data: Original	-
LS_PIN	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized	-
LS_PCA	P, Q, V^2, θ_{ref}	V, PF	$N_p \in \{40, 50, \cdots, 80\}$; $N_{cv} = 5$; Data: Normalized	-
LS_GEN	P, Q, V^2, θ_{ref}	V, PF	Covariance Estimation Model: ARIMA; Data: Normalized	FGLS[&]
PLS_NIP	P, Q, V^2, θ_{ref}	V, PF	Data: Normalized	-
PLS_CLS	P, Q, V^2, θ_{ref}	V, PF	$k \in \{2, 3, \cdots, 10\}$; $N_{cv} = 5$; Data: Normalized	-
DC	P	θ	Data: Original	-
PTDF	P	PF	Data: Original	-
TAY	P, Q	V, θ	Data: Original	-
DLPF	P, Q, V, θ_{ref}	V, PF	Data: Original	-

◇ : The default predictors and responses are $\{P,\ Q,\ V^2,\ \theta_{ref}\}$ and $\{V,\ \text{PF}\}$, respectively, unless they do not apply to the approach (see Table III).
¶ : The unit energy normalization is used for training and testing datasets by default, unless the approach is not applicable (see Table III).
† : N_{cv} is the fold number for cross-validation, used for auto-tuning the parameter based on the given parameter set and the training dataset.
‡ : “New Data Share” is the fraction of new data in the training set, used for recursive DPFL methods to update models in a point-wise manner.
§ : Every training data point has been used as a tangent point to find the min/max coefficients among all data points.
& : FGLS in MATLAB has been modified by replacing the ordinary least squares with the least squares with Pseudoinverse, as previously mentioned.

Comment 4.12: Limited Observability: Can the authors include references which have tackled this issue (if there is?) This problem is between state estimation and topology estimation.

Response to Comment 4.12: We are not aware of any existing studies in the data-driven power flow linearization literature that specifically address the challenge of limited observability. Therefore, we have included it as a future research direction in the revised manuscript.

Meanwhile, we appreciate the reviewer's insightful suggestion. While limited observability has not yet been directly tackled in DPFL research, this challenge has been well studied in the context of state and topology estimation. Following the reviewer's suggestion, we have expanded our discussion on this topic in the future challenges section. In particular, we have identified two potential solution paths: (i) employing existing state or topology estimation methods capable of handling limited observability to first reconstruct system states, which can then be used as inputs for data-driven linearization; or (ii) exploring joint optimization frameworks that simultaneously address both state/topology estimation and linearization.

This expanded discussion is now included in the future research directions section under the topic of limited observability, as given below:

Section VIII.3. Limited Observability

With partial system measurements, DPFL models are limited to observed variables, resulting in incomplete representations. Developing models that infer system-wide behavior from partial data remains an open and essential challenge, particularly in distribution grids. One potential path is to leverage existing state or topology estimation methods, which are specifically designed to address limited observability, and use their outputs to construct more complete DPFL models. Alternatively, a more integrated approach could involve jointly optimizing both state/topology estimation and power flow linearization within a unified framework, thereby directly addressing the observability challenge in the context of DPFL.

Comment 4.13: Minor comments/typos: Page 1/line 55: firstlyly, line 59 secondlyly, thirdlyly; Page 1/line 12: twice [7]; Page 6/line 20: PPDL, should it be PPFL? (also line 27); Page 9/line14: an approach.

Response to Comment 4.13: We have corrected all the specific errors mentioned by the reviewer. In addition, we have thoroughly proofread the entire manuscript to improve the language and presentation further using grammar-checking tools.

REFERENCES

[1] M. Jia and G. Hug, "Overview of data-driven power flow linearization," in *2023 IEEE Belgrade PowerTech*, 2023, pp. 01–06.

[2] H. Xu, A. D. Domínguez-García, V. V. Veeravalli, and P. W. Sauer, "Data-driven voltage regulation in radial power distribution systems," *IEEE Trans. on Power Systems*, vol. 35, no. 3, pp. 2133–2143, 2019.

[3] Z. Shao, Q. Zhai, J. Wu, and X. Guan, "Data based linear power flow model: Investigation of a least-squares based approximation," *IEEE Trans. on Power Systems*, vol. 36, no. 5, pp. 4246–4258, 2021.

[4] S. Hentong, Z. Qiaozhu, W. Jiang, and G. Xiaohong, "Data based linearization: Least-squares based approximation," *arXiv preprint arXiv:2007.02494*, 2020.

[5] Y. Liu, Z. Li, and Y. Zhou, "Data-driven-aided linear three-phase power flow model for distribution power systems," *IEEE Trans. on Power Systems*, vol. 37, no. 4, pp. 2783–2795, 2022.

[6] ——, "A physics-based and data-driven linear three-phase power flow model for distribution power systems," *arXiv preprint arXiv:2103.10147*, 2021.

[7] C. Mugnier, K. Christakou, J. Jaton, M. De Vivo, M. Carpita, and M. Paolone, "Model-less/measurement-based computation of voltage sensitivities in unbalanced electrical distribution networks," in *2016 Power Systems Computation Conference (PSCC)*. IEEE, 2016, pp. 1–7.

[8] Y. Liu, Y. Wang, N. Zhang, D. Lu, and C. Kang, "A data-driven approach to linearize power flow equations considering measurement noise," *IEEE Trans. on Smart Grid*, vol. 11, no. 3, pp. 2576–2587, 2019.

[9] Y. Liu, Z. Li, and S. Sun, "A data-driven method for online constructing linear power flow model," *IEEE Transactions on Industry Applications*, 2023.

[10] Y. Tan, Y. Chen, Y. Li, and Y. Cao, "Linearizing power flow model: A hybrid physical model-driven and data-driven approach," *IEEE Trans. on Power Systems*, vol. 35, no. 3, pp. 2475–2478, 2020.

[11] S. Nowak, Y. C. Chen, and L. Wang, "Measurement-based optimal der dispatch with a recursively estimated sensitivity model," *IEEE Trans. on Power Systems*, vol. 35, no. 6, pp. 4792–4802, 2020.

[12] Y. Chen, C. Wu, and J. Qi, "Data-driven power flow method based on exact linear regression equations," *Journal of Modern Power Systems and Clean Energy*, 2021.

[13] J. Zhang, Z. Wang, X. Zheng, L. Guan, and C. Chung, "Locally weighted ridge regression for power system online sensitivity identification considering data collinearity," *IEEE Trans. on Power Systems*, vol. 33, no. 2, pp. 1624–1634, 2017.

[14] J. Chen, W. Wu, and L. A. Roald, "Data-driven piecewise linearization for distribution three-phase stochastic power flow," *IEEE Trans. on Smart Grid*, 2021.

[15] J. Chen, W. Li, W. Wu, T. Zhu, Z. Wang, and C. Zhao, "Robust data-driven linearization for distribution three-phase power flow," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2020, pp. 1527–1532.

[16] J. Yu, Y. Weng, and R. Rajagopal, "Robust mapping rule estimation for power flow analysis in distribution grids," in *2017 North American Power Symposium (NAPS)*. IEEE, 2017, pp. 1–6.

[17] ——, "Mapping rule estimation for power flow analysis in distribution grids," *arXiv preprint arXiv:1702.07948*, 2017.

[18] Y. Liu, B. Xu, A. Botterud, N. Zhang, and C. Kang, "Bounding regression errors in data-driven power grid steady-state models," *IEEE Trans. on Power Systems*, vol. 36, no. 2, pp. 1023–1033, 2020.

[19] Z. Shao, Q. Zhai, Z. Han, and X. Guan, "A linear ac unit commitment formulation: An application of data-driven linear power flow model," *International Journal of Electrical Power & Energy Systems*, vol. 145, p. 108673, 2023.

[20] Y. Liu, Z. Li, and J. Zhao, "Robust data-driven linear power flow model with probability constrained worst-case errors," *arXiv preprint arXiv:2112.10320*, 2021.

[21] G. Yan and Z. Li, "Construction of an outlier-immune data-driven power flow model for model-absent distribution systems," *IEEE Transactions on Power Systems*, 2024.

[22] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, "Data-driven power flow linearization: A regression approach," *IEEE Trans. on Smart Grid*, vol. 10, no. 3, pp. 2569–2580, 2018.

[23] L. Guo, Y. Zhang, X. Li, Z. Wang, Y. Liu, L. Bai, and C. Wang, "Data-driven power flow calculation method: A lifting dimension linear regression approach," *IEEE Trans. on Power Systems*, vol. 37, no. 3, pp. 1798–1808, 2022.

[24] X. Li, "Fast heuristic ac power flow analysis with data-driven enhanced linearized model," *Energies*, vol. 13, no. 13, p. 3308, 2020.

[25] X. Li and K. Hedman, "Data driven linearized ac power flow model with regression analysis," *arXiv preprint arXiv:1811.09727*, 2018.

[26] M. Jia, Q. Cao, C. Shen, and G. Hug, "Frequency-control-aware probabilistic load flow: An analytical method," *IEEE Trans. on Power Systems*, pp. 1–16, 2022.

[27] M. Jia, W. Y. Chan, and G. Hug, "Daline: A data-driven power flow linearization toolbox for power systems research and education," 2024.

[28] M. Jia, G. Hug, N. Zhang, Z. Wang, Y. Wang, and C. Kang, "Supplementary material for generalized data-driven power flow linearization: Methodologies, evaluations, and challenges," 2025, available: https://github.com/JarvisETHZ/DPFL/blob/main/Supplementary%20Material.pdf.

[29] B. Taheri and D. K. Molzahn, "Optimizing parameters of the dc power flow," *Electric Power Systems Research*, vol. 235, p. 110719, 2024.

[30] J. Yang, N. Zhang, C. Kang, and Q. Xia, "A state-independent linear power flow model with accurate estimation of voltage magnitude," *IEEE Trans. on Power Systems*, vol. 32, no. 5, pp. 3607–3617, 2016.

[31] S. Nowak, Y. C. Chen, and L. Wang, "Distributed measurement-based optimal der dispatch with estimated sensitivity models," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 2197–2208, 2021.

[32] S. Talkington, S. Grijalva, M. J. Reno, J. A. Azzolini, and J. Peppanen, "Localized structure in secondary distribution system voltage sensitivity matrices," *Electric Power Systems Research*, vol. 226, p. 109788, 2024.

[33] M. Brown, M. Biswal, S. Brahma, S. J. Ranade, and H. Cao, "Characterizing and quantifying noise in pmu data," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.

[34] Y. Chen, F. Pan, F. Qiu, A. S. Xavier, T. Zheng, M. Marwali, B. Knueven, Y. Guan, P. B. Luh, L. Wu, B. Yan, M. A. Bragin, H. Zhong, A. Giacomoni, R. Baldick, B. Gisin, Q. Gu, R. Philbrick, and F. Li, "Security-constrained unit commitment for electricity market: Modeling, solution methods, and future challenges," *IEEE Trans. on Power Systems*, vol. 38, no. 5, pp. 4668–4681, 2023.

[35] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Foundations and Trends® in Electric Energy Systems*, vol. 4, no. 1-2, pp. 1–221, 2019.