Optimizing Parameters of the LinDistFlow Power Flow Approximation for Distribution Systems

Babak Taheri, Rahul K. Gupta, and Daniel K. Molzahn

***Abstract*—The DistFlow model accurately represents power flows in distribution systems, but the model's nonlinearities result in computational challenges for many applications. Accordingly, a linear approximation known as LinDistFlow (and its three-phase extension LinDist3Flow) is commonly employed. This paper introduces a parameter optimization algorithm for enhancing the accuracy of this approximation for both balanced single-phase equivalent and unbalanced three-phase distribution network models, with the goal of aligning the outputs more closely with those from the nonlinear DistFlow model. Using sensitivity information, our algorithm optimizes the LinDistFlow approximation's coefficient and bias parameters to minimize discrepancies in predictions of voltage magnitudes relative to the nonlinear DistFlow model. The parameter optimization algorithm employs the Truncated Newton Conjugate-Gradient (TNC) method to fine-tune coefficients and bias parameters during an offline training phase to improve the LinDistFlow approximation's accuracy. Numerical results underscore the algorithm's efficacy, showcasing accuracy improvements in L_1-norm and L_∞-norm losses of up to 92% and 88%, respectively, relative to the traditional LinDistFlow model. We also assess how the optimized parameters perform under changes in the network topology and demonstrate the optimized LinDistFlow approximation's efficacy in a hosting capacity optimization problem.**

***Index Terms*—DistFlow, LinDistFlow, machine learning, parameter optimization, distribution systems, hosting capacity.**

I. Introduction

Power flow models relating power injections, line flows, and voltages are central to power system design and operation [1]. While all mathematical models of physical systems are inherently approximations, the AC power flow equations are the established, high-fidelity benchmark used to accurately model these relationships. Incorporating the AC power flow equations into optimization problems introduces significant computational challenges due to these equations' nonlinearity, even for the radial networks of typical distribution systems [2]. These challenges are particularly relevant to problems that are run online to inform real-time decisions [3], [4], consider uncertainties [5], and model discrete decisions [6]–[9]. To address these challenges, engineers frequently use power flow approximations that trade accuracy for tractability [10].

Distribution systems are often modeled using an AC power flow formulation known as "DistFlow" [11]–[13]. A linearization of the DistFlow model known as "LinDistFlow" [11]–[13], or one of its variants [14]–[20], is commonly employed to make distribution system optimization problems tractable. An extension known as LinDist3Flow is often used for unbalanced three-phase distribution network models [14]. The traditional LinDistFlow approximation linearizes the DistFlow equations by assuming that the active and reactive line losses are much smaller than the active and reactive line flows. LinDistFlow has been used to site and size capacitors [11], [12], optimize the network topology [13], compute inverter setpoints and perform Volt/VAr control [15], [16], [21], [22], formulate stochastic problems [23], ensure fairness in solar photovoltaic curtailments [24], and calculate electricity prices [25], among many other applications. Despite their extensive applications, the accuracy of such models can vary, particularly outside near-nominal operating regions, as indicated in [16], [26]. Recent work continues to examine the inherent errors in the LinDistFlow model and their impact on applications such as flexibility aggregation, proposing compensation methods to refine its accuracy [27]. This variability in accuracy poses a significant challenge, as reliance on potentially inaccurate models can lead to suboptimal operational decisions, misjudgment of system limits (such as hosting capacity), and inefficient investment in distribution system infrastructure. Therefore, enhancing the fidelity of these linear approximations without compromising their computational tractability is of considerable practical importance for the reliable and economic operation and planning of modern distribution networks.

Building on ideas from recently developed "adaptive" power flow approximations [28]–[42], and related advancements focusing on areas like second-order sensitivity insights [43] and sample-based piecewise linear approaches [44], [45], this paper introduces a parameter optimization algorithm for the LinDistFlow model (and its three-phase extension, LinDist3Flow) to significantly enhance its accuracy. Our method optimizes key coefficient parameters, specifically the elements of the diagonal matrices D_r and D_x which represent effective line resistances and reactances, and newly introduced bias parameters. These bias parameters are vector offsets $\boldsymbol{\rho}$ for active power injections (p), $\boldsymbol{\varrho}$ for reactive power injections (q), and $\boldsymbol{\gamma}$ for squared voltage magnitudes (v). These parameters, which form the basis of our Optimized LinDistFlow (OLDF) formulation detailed further in Section III, are tuned via an offline training process. This allows the OLDF model to better emulate the nonlinear DistFlow model, thereby more accurately capturing the net impact of complex physical effects, such as those related to power losses, which are simplified in traditional LinDistFlow formulations.

Adaptive power flow approximations like the Optimized LinDistFlow in this paper are linearizations that are tailored to a specific system and operating range of interest. This contrasts with traditional power flow approximations that are often derived using general assumptions about broad classes

Electrical and Computer Engineering, Georgia Institute of Technology. {taheri, rahul.gupta, molzahn}@gatech.edu. Support from NSF #2145564.

of systems or are based on a particular nominal operating point. Adaptive power flow approximations invest computing time up front to calculate linearization coefficients in order to achieve increased accuracy when the approximations are deployed in an optimization problem. Thus, adaptive approximations are well suited for settings with both offline and online aspects (e.g., using a day-ahead forecast to compute linearization coefficients that are used for online computations in real-time applications [3], [4]) as well as settings where a nonlinear AC power flow model would lead to an intractable formulation [5]–[9].

Our prior work in [35] proposes an adaptive DC power flow approximation that optimizes parameters for a particular system over a specified operating range of interest. Other adaptive power flow approximations minimize worst-case error [28] and expected error [30] or leverage sample-based regression approaches [31]–[33] including techniques for constructing overestimating and underestimating approximations [34]. Recent advancements in this area explore the use of second-order sensitivities to both analyze and improve these approximations, guiding the development of, for instance, rational approximations and conservative piecewise linear functions tailored to directions of high curvature [43], as well as sample-based piecewise linear models [45]; see [36] for a recent survey.

Recently, there has been growing interest in developing advanced parameter optimization techniques for various linear power flow models. While our work focuses on the LinDistFlow approximation for distribution systems, several noteworthy approaches have also been proposed for transmission systems, often targeting the DC power flow model and its variants (e.g., [35], [39]–[42], [44]). These methodologies for transmission systems address different objectives and parameterization scopes, conditioned by the distinct topological and electrical characteristics of high-voltage networks. For instance, recent work by the authors has explored parameter optimization for improving DC Optimal Power Flow (DC-OPF) formulations [39] and for AC-informed DC optimal transmission switching problems [42] in such transmission system contexts. The continued application of LinDistFlow in distribution system analyses, such as in data-driven distributionally robust optimal power flow [46], underscores the ongoing need for enhancing its accuracy.

Contrasting with prior adaptive power flow approximations [28], [30]–[34], [36], this paper develops a parameter optimization algorithm that maintains the structure of the LinDistFlow approximation (and the unbalanced three-phase extension LinDist3Flow) as dictated by the network topology. Thus, the resulting parameter-optimized LinDistFlow approximation has the key advantages of being directly deployable in the many existing distribution system applications that rely on LinDistFlow (e.g., [11]–[13], [15], [16], [21]–[25]) and enabling straightforward modeling of topology changes.

While not using traditional machine learning models like neural networks, our parameter optimization algorithm draws inspiration from methods for training machine learning models. We define a loss function that compares the AC power flow solutions to the approximation's outputs over a set of sampled power injections. With analytically calculated parameter sensitivities, an offline training phase computes the parameter values that minimize this loss function using a Truncated Newton Conjugate-Gradient (TNC) method. Using linear programming or mixed-integer linear programming solvers, the optimized parameters are then used in online calculations for real-time settings or in problems for which a nonlinear AC power flow model would lead to intractability. The proposed approach is conceptually similar to our previous work for optimizing the parameters of the DC power flow approximation [35], but is applicable to unbalanced three-phase distribution systems.

Numerical comparisons demonstrate substantial accuracy advantages of our proposed algorithm compared to the traditional LinDistFlow/LinDist3Flow approximation as well as several recent LinDistFlow variants that also tune parameter values [17]–[19]. We also study how the approximation accuracy is affected by changes in network topology. Finally, to illustrate the benefits of our algorithm, we apply the optimized LinDistFlow parameters in a hosting capacity analysis.

To summarize, the key contributions of this paper are:

- **A systematic framework for optimizing LinDistFlow and LinDist3Flow** to significantly enhance accuracy to better match nonlinear DistFlow by tuning coefficient and bias parameters using sensitivity analysis and the TNC method.
- **A training methodology demonstrating enhanced model robustness** by optimizing LinDistFlow parameters over diverse operating conditions, outperforming traditional fixed-assumption or single-point methods.
- **Comprehensive evidence regarding the superior accuracy of the optimized LinDistFlow** (up to 92% L_1-norm and 88% L_∞-norm error reductions) over traditional and recent linear variants across diverse test systems, loading conditions, and for both single and three-phase models.
- **Insights into the optimized model's adaptability**, showcasing its ability to maintain enhanced accuracy across varying network topologies.
- **Demonstration of the proposed OLDF's efficacy in practical applications**, yielding more reliable and feasible solutions than traditional LDF in complex engineering tasks like hosting capacity analysis.

The remainder of this paper is organized as follows: Section II reviews the DistFlow model and LinDistFlow approximation. Section III introduces our proposed algorithm. Section IV extends the algorithm to three-phase network models. Section V presents numerical experiments that evaluate the performance of our algorithm. Section VI concludes the paper.

II. Power Flow Modeling

This section introduces the DistFlow formulation and its linear approximation, LinDistFlow. We focus on a balanced single-phase equivalent model to present the key concepts and then extend to unbalanced three-phase networks in Section IV.

A. Notation

We first establish notation. Let operators $(\cdot)^{-1}$, $|\cdot|$, $(\cdot)^\top$, and $(\cdot)^{-\top}$ denote the square matrix inverse, the absolute value of a number, the transpose of a matrix/vector, and the transpose

of a square matrix inverse, respectively. Let $\mathcal{N} := \{0, 1, \ldots, n\}$ and $\mathcal{E} := \{1, \ldots, n\}$ denote the sets of buses and lines, respectively, in a distribution network, where $|\mathcal{E}| = |\mathcal{N}| - 1$ for radial networks. Each bus $n \in \mathcal{N}$ has a voltage V_n. The set of all non-substation buses is denoted by $\mathcal{N}' := \{1, 2, ..., n\}$. Each bus $n \in \mathcal{N}'$ has a parent (upstream) bus denoted as π_n. Each line $(\pi_n, n) \in \mathcal{E}$ (connection between bus n and its parent π_n) has impedance $z_n = r_n + \mathrm{j}x_n$ with resistance r_n and reactance x_n, where $\mathrm{j} := \sqrt{-1}$ (note that lines are identified by their child buses to simplify notation; see Fig. 1). Let I_n and $S_n = P_n + \mathrm{j}Q_n$ denote the current and complex power flows, respectively, on line (π_n, n). The net power injection at bus n is $s_n = p_n + jq_n$. Additional variables include squared voltage and current magnitudes $v_n = |V_n|^2$ and $\ell_n = |I_n|^2$. $\mathbf{D}_r = \mathrm{diag}(\mathbf{r})$ and $\mathbf{D}_x = \mathrm{diag}(\mathbf{x})$ are diagonal matrices of resistances and reactances, and $\mathbf{p} = [p_1, \ldots, p_n]^\top$, $\mathbf{q} = [q_1, \ldots, q_n]^\top$, $\mathbf{P} = [P_1, \ldots, P_n]^\top$, and $\mathbf{Q} = [Q_1, \ldots, Q_n]^\top$ are vectors of active/reactive power injections and flows. Define $\hat{\mathbf{A}} = [\mathbf{a}_0 \ \mathbf{A}]$ as the $|\mathcal{E}| \times |\mathcal{N}|$ branch-bus incidence matrix describing the connections between the system's buses and branches, where $\mathbf{a}_0$ is the length-$|\mathcal{E}|$ vector associated with the substation bus and $\mathbf{A}$ is the reduced branch-bus matrix for all buses other than the substation bus. Let $\mathbf{v} := [|V_1|^2, \ldots, |V_n|^2]^\top$ represent squared voltage magnitudes and $v_0 = |V_0|^2$ correspond to the substation bus. To enhance voltage approximation accuracy, our algorithm also optimizes a set of bias parameters. These parameters are formulated as vectors of length $|\mathcal{N}'|$ and introduce corrective offsets to the model: γ is an offset for the squared voltage magnitudes (v), ρ is an offset for the active power injections (p), and ϱ is an offset for the reactive power injections (q).

B. DistFlow Model for Single-Phase Equivalent Networks

The DistFlow model [11]–[13] accurately represents voltage, current, and power flow relationships in radial distribution networks. The DistFlow model is:

$$v_n = |V_n|^2, \quad \forall n \in \mathcal{N}; \quad \ell_n = |I_n|^2, \qquad \forall n \in \mathcal{N}', \ \text{(1a)}$$

$$\sum_{k:n\to k} P_k = p_n + P_n - r_n\ell_n, \qquad \forall n \in \mathcal{N}', \ \text{(1b)}$$

$$\sum_{k:n\to k} Q_k = q_n + Q_n - x_n\ell_n, \qquad \forall n \in \mathcal{N}', \ \text{(1c)}$$

$$v_n = v_{\pi_n} - 2(r_nP_n + x_nQ_n) + (r_n^2 + x_n^2)\ell_n, \forall n \in \mathcal{N}', \ \text{(1d)}$$

$$v_{\pi_n}\ell_n = P_n^2 + Q_n^2, \qquad \forall n \in \mathcal{N}'. \ \text{(1e)}$$

Equation (1a) defines variables for the squared voltage magnitudes v_n and squared current flow magnitudes ℓ_n, which are used in place of the voltage phasors V_n and current phasors I_n. Equations (1b) and (1c) correspond to active and reactive power balance at each bus. Equation (1d) models the change in squared voltage magnitudes across lines. Equation (1e) uses the definition of apparent power to relate the squared voltages v_n, squared currents ℓ_n, and squared apparent power $P_n^2 + Q_n^2$. The DistFlow model is nonlinear due to (1e). Note that the DistFlow model (1) is equivalent to bus-injection AC power flow models for radial systems [11]–[13], [47], [48].

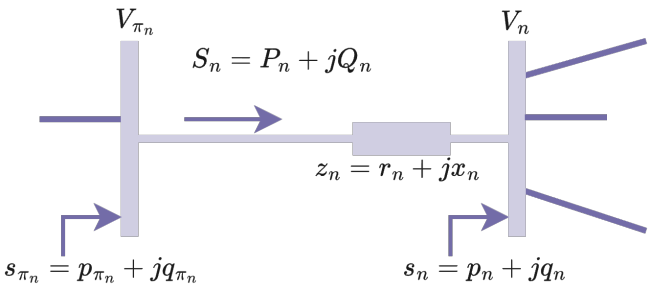


Fig. 1. A 2-bus system with line (π_n, n) feeding bus n from its parent π_n.

C. Traditional LinDistFlow Approximation

The LinDistFlow approximation linearizes the DistFlow equations by neglecting the line loss terms $r_n\ell_n$ in (1b), $x_n\ell_n$ in (1c), and $(r_n^2 + x_n^2)\ell_n$ in (1d) [11]–[13]. Without these, the nonlinear equation (1e) can be dropped, with the remaining equations linearly relating the squared voltage magnitudes, the power injections, and the line flows:

$$\sum_{k:n\to k} P_k \approx p_n + P_n, \qquad \forall n \in \mathcal{N}', \qquad \text{(2a)}$$

$$\sum_{k:n\to k} Q_k \approx q_n + Q_n, \qquad \forall n \in \mathcal{N}', \qquad \text{(2b)}$$

$$v_n \approx v_{\pi_n} - 2(r_nP_n + x_nQ_n), \qquad \forall n \in \mathcal{N}'. \qquad \text{(2c)}$$

A matrix form of the LinDistFlow approximation is:

$$\mathbf{D}_r = \mathrm{diag}(\mathbf{r}), \quad \mathbf{D}_x = \mathrm{diag}(\mathbf{x}), \qquad \text{(3a)}$$

$$\mathbf{p} = \mathbf{A}^\top\mathbf{P}, \qquad \text{(3b)}$$

$$\mathbf{q} = \mathbf{A}^\top\mathbf{Q}, \qquad \text{(3c)}$$

$$\mathbf{A}\mathbf{v} + v_0\mathbf{a}_0 = 2\mathbf{D}_r\mathbf{P} + 2\mathbf{D}_x\mathbf{Q}, \qquad \text{(3d)}$$

$$\mathbf{v} = v_0\mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r\mathbf{A}^{-\top}\mathbf{p} + 2\mathbf{A}^{-1}\mathbf{D}_x\mathbf{A}^{-\top}\mathbf{q}. \qquad \text{(3e)}$$

For radial networks where all lines have positive resistance ($r_n \geq 0$) and reactance ($x_n \geq 0$), the LinDistFlow approximation overestimates the voltage magnitudes and underestimates the complex power flows required to supply the loads [47]. The following section presents an algorithm to reduce this approximation error by optimizing the LinDistFlow parameters.

III. Optimized LinDistFlow Approximation (OLDF)

While the traditional LinDistFlow model derives the matrices D_r and D_x directly from the physical line parameters, our approach treats them as optimizable coefficients. The justification for this is that the LinDistFlow model is an *approximation* of the true nonlinear physics, primarily by neglecting the impacts of line losses. Since the model's structure is simplified, using the exact physical parameters does not guarantee the most accurate linear approximation. Instead, by optimally selecting values for the $\mathbf{D}_r$ and $\mathbf{D}_x$ coefficients, we find the "effective" resistance and reactance values that cause the simplified linear model to best match the results of the full nonlinear model over a range of operating conditions. We note that conceptually similar approaches for

selecting parameter values in power flow approximations are employed in, e.g., [19], [28]–[45]. In Section V, we numerically compare our optimized LinDistFlow formulation to the relevant alternative LinDistFlow approximations to show that optimizing the parameter values is an effective method for improving LinDistFlow approximation accuracy.

We first generalize the LinDistFlow approximation by introducing bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ that offset the squared voltage magnitudes and active and reactive power injections:

$$\mathbf{v} = v_0\mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r\mathbf{A}^{-\top}(\mathbf{p}+\boldsymbol{\rho}) + 2\mathbf{A}^{-1}\mathbf{D}_x\mathbf{A}^{-\top}(\mathbf{q}+\boldsymbol{\varrho}) + \boldsymbol{\gamma}. \quad (4)$$

Appropriate selection of parameter values for $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ can significantly improve the LinDistFlow approximation's accuracy. By adjusting the effective nodal power injections $(\mathbf{p}+\boldsymbol{\rho})$ and $(\mathbf{q}+\boldsymbol{\varrho})$, the parameters $\boldsymbol{\rho}$ and $\boldsymbol{\varrho}$ allow the linear model to implicitly account for the voltage impact of system characteristics not directly captured in the basic LinDistFlow equations, such as line losses.[1] We next introduce a machine learning-inspired algorithm which optimizes these parameters to reduce the discrepancy between the voltages output by the LinDistFlow approximation (4) and the DistFlow model (1).

As shown in Fig. 2, we propose a two-phase algorithm for parameter optimization with an *offline* training phase followed by an *online* application phase. The *offline* phase, conducted once, optimizes the values for the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$. This initial computation seeks to align the LinDistFlow approximation with the DistFlow model's behavior across varied operational scenarios. The optimized parameters are then used in the *online* phase to improve accuracy in various applications while enabling tractable solution with linear programming and mixed-integer linear programming solvers.

To optimize the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$, Section III-A first formulates a loss function that gauges the approximation's accuracy by comparing its voltage predictions to those from the DistFlow model across sampled operational conditions. Section III-B then analytically derives the loss function's sensitivities with respect to the coefficient and bias parameters. Informed by these sensitivities, Section III-C applies the TNC optimization method to minimize the loss function. The resulting optimized parameters are used online for various applications, as benchmarked in Section V.

A. Formulation of the Loss Function

We define a loss function, $\mathcal{L}$, as the mean square error between the voltage solutions of the DistFlow model ($\mathbf{v}_m^{DF}$) and our optimized LinDistFlow approximation ($\mathbf{v}_m^{OLDF}$) across a given set of load scenarios $\mathcal{M} = \{1, 2, \ldots, S\}$:

$$\mathcal{L}(\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}) = \frac{1}{|\mathcal{M}||\mathcal{N}'|}\sum_{m\in\mathcal{M}} ||\mathbf{v}_m^{OLDF} - \mathbf{v}_m^{DF}||_2^2,$$
$$= \frac{1}{|\mathcal{M}||\mathcal{N}'|}\sum_{m\in\mathcal{M}} ||v_0\mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r\mathbf{A}^{-\top}(\mathbf{p}+\boldsymbol{\rho}) + 2\mathbf{A}^{-1}\mathbf{D}_x\mathbf{A}^{-\top}(\mathbf{q}+\boldsymbol{\varrho}) + \boldsymbol{\gamma} - \mathbf{v}_m^{DF}||_2^2, \quad (5)$$

where normalization by $\frac{1}{|\mathcal{M}||\mathcal{N}'|}$ adjusts for the system size and number of samples. Using mean square error ensures that larger deviations are more heavily penalized, aligning with typical operational priorities where minimizing the most significant errors is often more important than reducing numerous smaller inaccuracies.

The optimal LinDistFlow parameters minimize $\mathcal{L}$:

$$\underset{\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}}{\text{minimize}} \ \mathcal{L}(\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}). \quad (6)$$

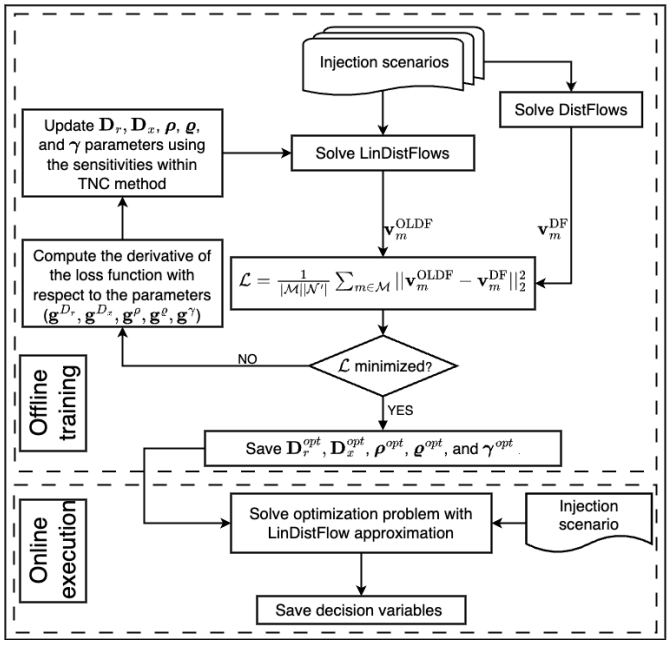


Fig. 2. Flowchart depicting the proposed parameter optimization algorithm.

B. Parameter Sensitivity Analysis

We use the TNC optimization method to solve (6). This method relies on the gradients of the loss function with respect to the parameters, which we present next. We start with the sensitivities for the parameters, represented generically by $\mathbf{g}^{\eta}$, where η can be any of the parameter sets $\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}$, and $\boldsymbol{\varrho}$. These sensitivities are calculated as follows:

$$\mathbf{g}^{\eta} = \frac{2}{|\mathcal{M}||\mathcal{N}'|}\sum_{m\in\mathcal{M}}\Big(\frac{\partial \mathbf{v}^{OLDF}}{\partial \eta}\Big|_{\mathbf{v}_m^{OLDF}}\Big)^{\top}\Big(\mathbf{v}_m^{OLDF} - \mathbf{v}_m^{DF}\Big), \quad (7a)$$

where $\eta \in \{\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}\}$. The partial derivatives are:

$$\frac{\partial \mathbf{v}^{OLDF}}{\partial \mathbf{D}_r} = 2\mathbf{A}^{-1}\text{diag}\Big(\mathbf{A}^{-\top}(\mathbf{p}+\boldsymbol{\rho})\Big), \quad (7b)$$
$$\frac{\partial \mathbf{v}^{OLDF}}{\partial \mathbf{D}_x} = 2\mathbf{A}^{-1}\text{diag}\Big(\mathbf{A}^{-\top}(\mathbf{q}+\boldsymbol{\varrho})\Big), \quad (7c)$$
$$\frac{\partial \mathbf{v}^{OLDF}}{\partial \boldsymbol{\gamma}} = \mathbf{I}, \quad (7d)$$
$$\frac{\partial \mathbf{v}^{OLDF}}{\partial \boldsymbol{\rho}} = 2\mathbf{A}^{-1}\mathbf{D}_r\mathbf{A}^{-\top}, \quad (7e)$$
$$\frac{\partial \mathbf{v}^{OLDF}}{\partial \boldsymbol{\varrho}} = 2\mathbf{A}^{-1}\mathbf{D}_x\mathbf{A}^{-\top}. \quad (7f)$$

[1] Similar adjustment parameters are often employed to improve the accuracy of DC power flow approximations [49].

Here, $\mathbf{I}$ is the identity matrix. These sensitivities enable gradient-based methods such as TNC for optimizing the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$, as we will describe next.

C. Implementation of the Optimization Solution

The gradients in Section III-B enable the application of gradient-based optimization methods such as TNC [50], [51] to solve the parameter optimization problem (6). The choice of TNC as opposed to other gradient-based optimization methods such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) and limited-memory BFGS (L-BFGS) [50], [51] is based on the TNC method's superior scalability in our empirical testing. Specifically, for our parameter optimization problem, TNC demonstrated superior practical performance over L-BFGS variants, converging with less computational time, particularly for larger test cases. This advantage in handling large-scale systems is consistent with our findings in related work [35], where TNC also proved more effective.

As illustrated in Algorithm 1, TNC iteratively uses Hessian-vector products to approximate the Newton direction, effectively managing memory and computation even in high-dimensional spaces. TNC operates by approximating the solution to Newton's equations for a function's local minimum, truncating early to conserve computational resources while still moving significantly towards the minimum. This is particularly advantageous for problems where evaluating the full Hessian matrix is difficult. To compute (6) in order to find the optimal LinDistFlow parameters, we utilize SciPy's `scipy.optimize.minimize` TNC implementation with the sensitivities from Section III-B, using gradient norm thresholds and iteration limits as termination criteria. SciPy's TNC internally computes the required Hessian-vector products ($\mathbf{Hp}$ in Algorithm 1) using finite differences of the analytical gradients from Section III-B, as no custom Hessian-vector product function was supplied. We employed default SciPy TNC values for the Wolfe line search parameters (α_1 and α_2 in Algorithm 1) and applied no custom preconditioner, effectively treating $\mathbf{M}$ in Algorithm 1 as the identity matrix.

Algorithm 1: Truncated Newton (TNC) Method

Input: $\mathbf{x}_0 = [\mathbf{D}_r^\top, \mathbf{D}_x^\top, \boldsymbol{\gamma}^\top, \boldsymbol{\rho}^\top, \boldsymbol{\varrho}^\top]^\top$: Initial guess
ϵ: Tolerance for convergence
max_iter: Maximum iterations
$\mathcal{L}(\mathbf{x}_k)$: Loss function
$\nabla\mathcal{L}(\mathbf{x}_k)= \mathbf{g} = [\mathbf{g}^{D_r\top}, \mathbf{g}^{D_x\top}, \mathbf{g}^{\gamma\top}, \mathbf{g}^{\rho\top}, \mathbf{g}^{\varrho\top}]^\top$
$\mathbf{M}$: Preconditioning matrix (often a diagonal matrix)
$\mathbf{H}$: Hessian or its approximation function
α_1: Armijo condition constant, small (e.g., 10^{-4})
α_2: Curvature condition constant, between α_1 and 1
Output: Optimized parameters $\mathbf{x}^*$

Initialize $\mathbf{x}_k \leftarrow \mathbf{x}_0$
$k \leftarrow 0$
while $k \leq$ *max_iter and* $\|\nabla\mathcal{L}(\mathbf{x}_k)\| > \epsilon$ **do**
  $\mathbf{g} \leftarrow \nabla\mathcal{L}(\mathbf{x}_k)$ `// Compute the gradient at` $\mathbf{x}_k$
  $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{g}$ `// Apply the preconditioner`
  $\mathbf{r} \leftarrow -\mathbf{g}$, $\mathbf{p} \leftarrow \mathbf{z}$
  $\rho_{old} \leftarrow \mathbf{r}^\top\mathbf{z}$
  while $\|\mathbf{r}\| > \epsilon$ **do**
    $\mathbf{q} \leftarrow \mathbf{Hp}$ `// Hessian-vector product`
    $\alpha \leftarrow \frac{\rho_{old}}{\mathbf{p}^\top\mathbf{q}}$
    $\mathbf{p} \leftarrow \mathbf{z} + \eta\mathbf{p}$
    $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{q}$
    $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$
    $\rho_{new} \leftarrow \mathbf{r}^\top\mathbf{z}$
    $\eta \leftarrow \frac{\rho_{new}}{\rho_{old}}$
    $\rho_{old} \leftarrow \rho_{new}$
  `// Wolfe Line Search to determine` β
  $\beta \leftarrow 1$ `// Initial step length`
  while *True* **do**
    if $\mathcal{L}(\mathbf{x}_k + \beta\mathbf{p}) \leq \mathcal{L}(\mathbf{x}_k) + \alpha_1\beta\mathbf{g}^\top\mathbf{p}$ ***and*** $\|\nabla\mathcal{L}(\mathbf{x}_k + \beta\mathbf{p})^\top\mathbf{p}\| \leq \alpha_2\|\mathbf{g}^\top\mathbf{p}\|$ **then**
      Break
    $\beta \leftarrow \beta/2$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \beta\mathbf{p}$ `// Update the parameter vector`
  $k \leftarrow k + 1$
$\mathbf{x}^* \leftarrow \mathbf{x}_k$

IV. Three-Phase LinDistFlow (LinDist3Flow)

We next discuss extensions to unbalanced three-phase distribution systems [14]. The equivalent of (4) for LinDist3Flow is (see Appendix A for details):

$$\mathbb{V} = \mathbb{V}_0\mathbf{1} + \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^P)\mathbb{A}_3^{-\top}(\mathbb{P} + \boldsymbol{\rho}_3) + \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^Q)\mathbb{A}_3^{-\top}(\mathbb{Q} + \boldsymbol{\varrho}_3) + \boldsymbol{\gamma}_3. \quad (8)$$

where $\mathbb{V} = \begin{bmatrix} v_1^a & v_1^b & v_1^c & \cdots & v_n^a & v_n^b & v_n^c \end{bmatrix}^\top$ is the vector of squared voltage magnitudes for each phase (a, b, c), $\mathbb{A}_3$ is the network incidence matrix, $\text{bdiag}(\cdot)$ is the block diagonal operator, $\mathbb{P} = \begin{bmatrix} p_1^a & p_1^b & p_1^c & \cdots & p_n^a & p_n^b & p_n^c \end{bmatrix}^\top$ and $\mathbb{Q} = \begin{bmatrix} q_1^a & q_1^b & q_1^c & \cdots & q_n^a & q_n^b & q_n^c \end{bmatrix}^\top$ are the active and reactive power injection vectors, and the $\mathbb{H}$ matrices for each line $(i, j) \in \mathcal{E}$ are:

$$\mathbb{H}_{ij}^P = \begin{bmatrix} -2r_{ij}^{aa} & r_{ij}^{ab} - \sqrt{3}x_{ij}^{ab} & r_{ij}^{ac} + \sqrt{3}x_{ij}^{ac} \\ r_{ij}^{ba} + \sqrt{3}x_{ij}^{ba} & -2r_{ij}^{bb} & r_{ij}^{bc} - \sqrt{3}x_{ij}^{bc} \\ r_{ij}^{ca} - \sqrt{3}x_{ij}^{ca} & r_{ij}^{cb} + \sqrt{3}x_{ij}^{cb} & -2r_{ij}^{cc} \end{bmatrix}, \quad (9)$$

$$\mathbb{H}_{ij}^Q = \begin{bmatrix} -2x_{ij}^{aa} & x_{ij}^{ab} + \sqrt{3}r_{ij}^{ab} & x_{ij}^{ac} - \sqrt{3}r_{ij}^{ac} \\ x_{ij}^{ba} - \sqrt{3}r_{ij}^{ba} & -2x_{ij}^{bb} & x_{ij}^{bc} + \sqrt{3}r_{ij}^{bc} \\ x_{ij}^{ca} + \sqrt{3}r_{ij}^{ca} & x_{ij}^{cb} - \sqrt{3}r_{ij}^{cb} & -2x_{ij}^{cc} \end{bmatrix}. \quad (10)$$

Analogous to the single-phase algorithm, we train the parameters $\mathbb{H}_{ij}^P$, $\mathbb{H}_{ij}^Q$, $\boldsymbol{\rho}_3$, $\boldsymbol{\varrho}_3$, and $\boldsymbol{\gamma}_3$ for three-phase networks. The parameters $\boldsymbol{\rho}_3$, $\boldsymbol{\varrho}_3$, and $\boldsymbol{\gamma}_3$ are the three-phase versions of $\boldsymbol{\rho}$, $\boldsymbol{\varrho}$, and $\boldsymbol{\gamma}$, respectively. To this end, we rewrite the loss function (5) as:

$$\begin{aligned}\mathcal{L}(\mathbb{H}^P, \mathbb{H}^Q, \boldsymbol{\gamma}_3, \boldsymbol{\rho}_3, \boldsymbol{\varrho}_3) &= \frac{1}{|\mathcal{M}||3\mathcal{N}'|}\sum_{m\in\mathcal{M}} ||\mathbb{V}_m^{OLDF} - \mathbb{V}_m^{DF}||_2^2, \\ &= \frac{1}{|\mathcal{M}||3\mathcal{N}'|}\sum_{m\in\mathcal{M}} ||\mathbb{V}_0\mathbf{1} + \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^P)\mathbb{A}_3^{-\top}(\mathbb{P} + \boldsymbol{\rho}_3) + \\ &\quad \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^Q)\mathbb{A}_3^{-\top}(\mathbb{Q} + \boldsymbol{\varrho}_3) + \boldsymbol{\gamma}_3 - \mathbb{V}_m^{DF}||_2^2. \quad (11)\end{aligned}$$

The sensitivity of the loss function (11) with respect to the $\mathbb{H}_{ij}^P$, $\mathbb{H}_{ij}^Q$, $\boldsymbol{\rho}_3$, $\boldsymbol{\varrho}_3$, and $\boldsymbol{\gamma}_3$ parameters, necessary for gradient-based optimization methods such as TNC, can be computed similarly to the single-phase case as in (7). This yields:

$$\mathbf{g}^\eta = \frac{2}{|\mathcal{M}||3\mathcal{N}'|}\sum_{m\in\mathcal{M}}\Big(\frac{\partial\mathbb{V}^{OLDF}}{\partial\eta}\Big|_{\mathbb{V}_m^{OLDF}}\Big)^\top\Big(\mathbb{V}_m^{OLDF} - \mathbb{V}_m^{DF}\Big), \quad (12a)$$

where $\eta \in \{\text{bdiag}(\mathbb{H}^P), \text{bdiag}(\mathbb{H}^Q), \boldsymbol{\gamma}_3, \boldsymbol{\rho}_3, \boldsymbol{\varrho}_3\}$, i.e., the parameters to be optimized. The partial derivatives are:

$$\frac{\partial \mathbb{V}^{OLDF}}{\partial \text{bdiag}(\mathbb{H}^P)} = \left(\mathbb{A}_3^{-\top}(\mathbb{P} + \boldsymbol{\rho}_3)\right)^\top \otimes \mathbb{A}_3^{-1}, \tag{12b}$$

$$\frac{\partial \mathbb{V}^{OLDF}}{\partial \text{bdiag}(\mathbb{H}^Q)} = \left(\mathbb{A}_3^{-\top}(\mathbb{Q} + \boldsymbol{\varrho}_3)\right)^\top \otimes \mathbb{A}_3^{-1}, \tag{12c}$$

$$\frac{\partial \mathbb{V}^{OLDF}}{\partial \boldsymbol{\gamma}_3} = \mathbf{I}, \tag{12d}$$

$$\frac{\partial \mathbb{V}^{OLDF}}{\partial \boldsymbol{\rho}_3} = \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^P)\mathbb{A}_3^{-\top}, \tag{12e}$$

$$\frac{\partial \mathbb{V}^{OLDF}}{\partial \boldsymbol{\varrho}_3} = \mathbb{A}_3^{-1}\text{bdiag}(\mathbb{H}^Q)\mathbb{A}_3^{-\top}. \tag{12f}$$

Here, $\otimes$ denotes the Kronecker product.

V. Numerical Analysis

This section empirically benchmarks the proposed algorithm "optimized LinDistFlow" (OLDF) against several other related power flow linearizations. Specifically, in addition to the nonlinear DistFlow model (1) that provides the ground truth via an AC power flow solution, we also benchmark our optimized LinDistFlow against the traditional LinDistFlow approximation (LDF) [11]–[13], the parameterized linear power flow (PLPF) approximation from [19], the Lossy DistFlow (LoDF) approximation from [18], and the decoupled linear power flow (DLPF) approximation from [17]. We use various test distribution systems and loading scenarios to replicate the benchmarking methodologies adopted in this literature. These include the balanced single-phase equivalent test cases `IEEE 33-bus`, `IEEE 69-bus`, and modified `IEEE 123-bus` from [52] as well as the `22-bus`, `85-bus`, `141-bus`, `case533mt-hi`, and `case-eu906` systems from MATPOWER [53]. For three-phase distribution networks, we use the `IEEE 13-bus`, `IEEE 37-bus`, and `IEEE 123-bus` test cases.

A. Algorithm Training

As in [19], we use 20 power injection scenarios during our algorithm's training phase. This selection is supported by a sensitivity analysis (see Fig. 3), which showed diminishing returns in accuracy improvement with a larger number of scenarios, yielding a balance between model fidelity and training efficiency. These scenarios scale the nominal power injections at each bus by normally distributed multipliers with a mean of one and standard deviation of 35%. For the nonlinear DistFlow solutions, we use PowerModels.jl [54] and OpenDSS with the OpenDSSDirect.py package for single- and three-phase networks, respectively, on a PACE computing node at Georgia Tech with a 24-core CPU and 32 GB of RAM. The training algorithm is implemented in Python 3 in a Jupyter Notebook using the TNC method from scipy.optimize.minimize with objective function (5), Jacobian $\mathbf{g} = [\mathbf{g}^{D_r\top}, \mathbf{g}^{D_x\top}, \mathbf{g}^{\gamma\top}, \mathbf{g}^{\rho\top}, \mathbf{g}^{\varrho\top}]^\top$, convergence tolerance of 1×10^{-6} per unit, and iteration limit of 100.[2]

[2]Code is available at https://github.com/BabakTaheri1/OLDF

B. Performance Metrics

We quantify approximation accuracy by comparing the voltage magnitude outputs (denoted $\mathbf{v}^{[\text{model}]}$, where [model] is OLDF, PLPF, LoDF, LDF, or DLPF for the various approximations) against the nonlinear DistFlow solutions (denoted $\mathbf{v}^{DF}$). We use max and mean error metrics in per unit (p.u.):

$$\varepsilon_{\text{max}}^{[\text{model}]} = \max_{m \in \mathcal{M}} \|\mathbf{v}_m^{[\text{model}]} - \mathbf{v}_m^{\text{DF}}\|_\infty \tag{13}$$

$$\varepsilon_{\text{avg}}^{[\text{model}]} = \frac{1}{|\mathcal{M}||\mathcal{N}'|} \sum_{m \in \mathcal{M}} \|\mathbf{v}_m^{[\text{model}]} - \mathbf{v}_m^{\text{DF}}\|_1 \tag{14}$$

where $|\mathcal{M}|$ is the number of testing samples, $|\mathcal{N}'|$ is the number of non-root nodes in the distribution systems, $\|\cdot\|_\infty$ is the L_∞-norm, and $\|\cdot\|_1$ is the L_1-norm.

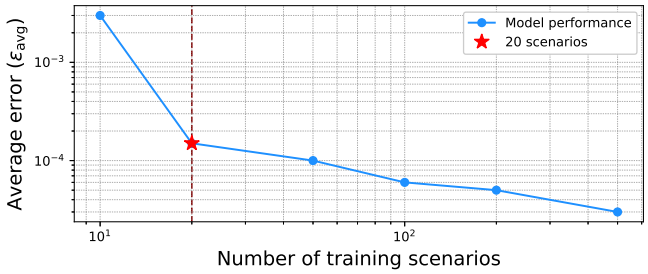


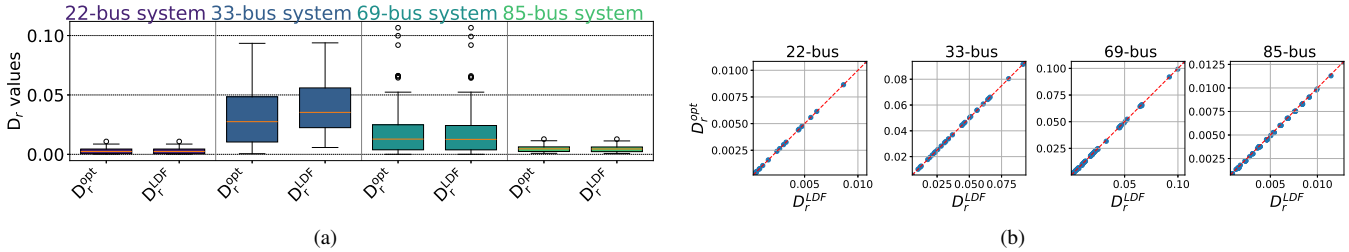
Fig. 3. Average error ($\epsilon_{\text{avg}}^{OLDF}$) as a function of the number of training scenarios for the `IEEE 33-bus` test system.

C. Parameter Optimization Analysis

We next present the results of the parameter optimization across various test cases by plotting the parameter values from the traditional LinDistFlow approximation and the LinDistFlow with optimized parameter values.

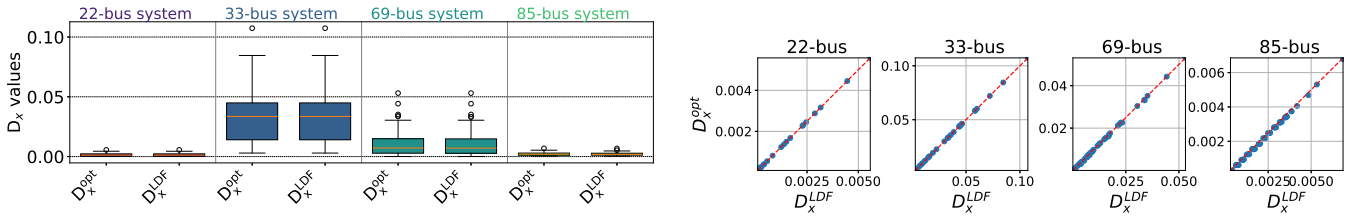
The box plots in Figs. 4 and 5 show the distributions of $\mathbf{D}_r$ and $\mathbf{D}_x$ parameter values, respectively, for the traditional LinDistFlow and our optimized parameters. Each box plot captures the interquartile range (IQR) with a median line. The whiskers extend to 1.5 times the IQR, with outliers represented as individual points. The horizontal lines at the whiskers' ends indicate the 90$^{\text{th}}$ percentile. For each test case, the box plots display two distributions: the LinDistFlow ($\mathbf{D}_r^{LDF}$ or $\mathbf{D}_x^{LDF}$) and the results from our optimization algorithm ($\mathbf{D}_r^{opt}$ and $\mathbf{D}_x^{opt}$). The optimized parameter values' distributions align closely with those from existing heuristics for $\mathbf{D}_r$ and $\mathbf{D}_x$. This indicates that our algorithm yields parameter values in a reasonable range, consistent with conventional heuristics.

Additionally, scatter plots accompanying these box plots compare LinDistFlow ($\mathbf{D}_r^{LDF}$, $\mathbf{D}_x^{LDF}$) and optimized ($\mathbf{D}_r^{opt}$, $\mathbf{D}_x^{opt}$) parameter values. The red dashed line at 45° in each subplot signifies a one-to-one correlation in the parameter values. These plots also show that the optimized parameters are broadly similar to those from existing heuristics, thus aligning with longstanding power engineering intuition that the line resistances and reactances are key parameters in dictating power flows. Despite the overall consistency with traditional LinDistFlow parameter choices, our results show that the optimized parameters yield significant accuracy improvements.



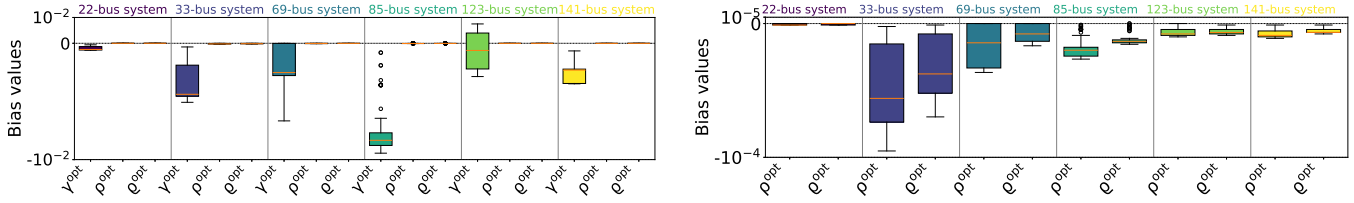
(a) (b)

Fig. 4. (a) Boxplots showing the distributions of the $\mathbf{D}_r$ parameter values for multiple test cases. Each test case is represented by two boxplots indicating the traditional and optimal $\mathbf{D}_r$ parameter values. (b) Scatter plots comparing the coefficient values $\mathbf{D}_r^{LDF}$ and $\mathbf{D}_r^{opt}$ for various test cases.



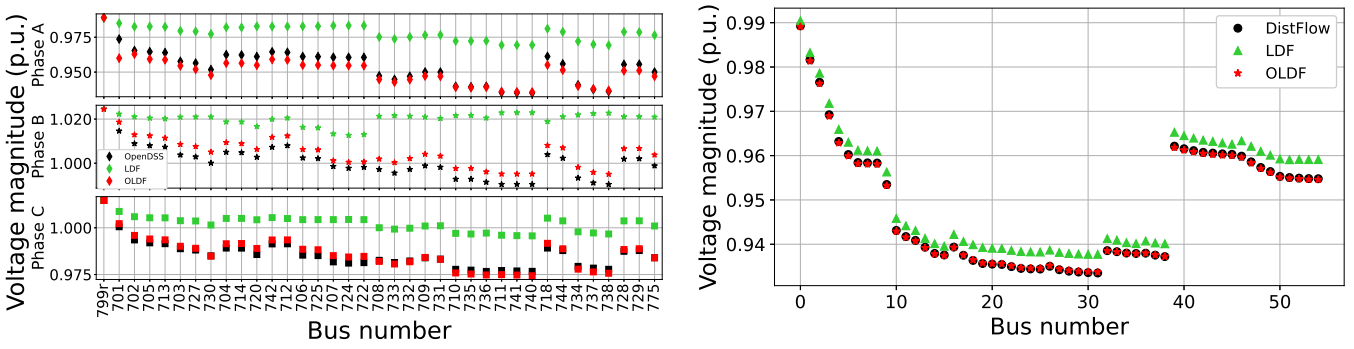
(a) (b)

Fig. 5. (a) Boxplots showing the distributions of the $\mathbf{D}_x$ parameter values for multiple test cases. Each test case is represented by two boxplots indicating the traditional and optimal $\mathbf{D}_x$ parameter values. (b) Scatter plots comparing the coefficient values $\mathbf{D}_x^{LDF}$ and $\mathbf{D}_x^{opt}$ for various test cases.



(a) (b)

Fig. 6. Boxplots showing the distributions of the bias parameters for multiple test cases. Each test case is represented by three boxplots indicating the $\boldsymbol{\gamma}^{opt}$, $\boldsymbol{\rho}^{opt}$, and $\boldsymbol{\varrho}^{opt}$ parameter values. (b) Plotting only $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ parameters for better comparison.



(a) `37-bus` (b) Modified `123-bus`

Fig. 7. Voltage profiles for the base case using the nonlinear DistFlow (OpenDSS in black), LinDistFlow with initial parameters (LDF in green), and proposed OLDF (LinDistFlow with optimized parameters in red).

The box plots in Fig. 6 show the distribution of the bias parameters. Illustrated in Fig. 6a, the optimal values of $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ are notably smaller in magnitude when compared to those of $\boldsymbol{\gamma}^{opt}$. Consequently, Fig. 6b presents the $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ parameters separately to highlight their distinct distributions. This disparity suggests simplifying the model by focusing on optimizing only three parameters: $\mathbf{D}_r$, $\mathbf{D}_x$, and $\boldsymbol{\gamma}$. However, our experimental results indicate that this simplification performs poorly for high-load scenarios. We therefore recommend optimizing all of the parameters.

D. Algorithm Testing

We next characterize the optimized parameter values' performance relative to alternative LinDistFlow formulations un-

TABLE I
MODEL EVALUATION - BASE LOAD

	Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{DLPF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{DLPF}$	$\varepsilon_{\max}^{OLDF}$
1ϕ	22-bus	0.00023	0.00014	0.00236	0.00040	**0.00001**	0.00030	0.00025	0.00314	0.00066	**0.00001**
	33-bus	0.00198	0.00080	0.00288	0.00368	**0.00015**	0.00284	0.00125	0.00402	0.00638	**0.00025**
	69-bus	0.00119	0.00075	0.00112	0.00186	**0.00023**	0.00388	0.00290	0.00327	0.00766	**0.00094**
	85-bus	0.00531	0.00180	0.00206	0.00942	**0.00056**	0.00663	0.00221	0.00261	0.01377	**0.00075**
	123-bus	0.00218	0.00160	0.00494	0.00348	**0.00018**	0.00255	0.00186	0.00579	0.00460	**0.00034**
	141-bus	0.00152	0.00071	0.00326	0.00280	**0.00003**	0.00207	0.00099	0.00409	0.00453	**0.00004**
	533-bus	0.00031	-	-	-	**0.00001**	0.00079	-	-	-	**0.00005**
	906-bus	0.02552	-	-	-	**0.00037**	0.02566	-	-	-	**0.00119**
3ϕ	13-bus	0.02642	-	-	-	**0.00901**	0.05899	-	-	-	**0.03558**
	37-bus	0.02022	-	-	-	**0.00325**	0.03435	-	-	-	**0.00710**
	123-bus	0.01388	-	-	-	**0.00340**	0.03707	-	-	-	**0.01322**

1ϕ stands for balanced single-phase equivalent networks, and 3ϕ stands for unbalanced three-phase networks.
The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

TABLE II
MODEL EVALUATION - HIGH LOAD

	Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{DLPF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{DLPF}$	$\varepsilon_{\max}^{OLDF}$
1ϕ	22-bus	0.00050	**0.00020**	0.00376	0.00091	0.00028	0.00132	**0.00080**	0.00733	0.00280	0.00097
	33-bus	0.00418	0.00239	0.00556	0.00795	**0.00224**	0.01573	**0.01018**	0.01713	0.03133	0.01254
	69-bus	0.00254	0.00199	0.00278	0.00403	**0.00129**	0.02253	0.01975	0.01926	0.03929	**0.01572**
	85-bus	0.01156	0.00817	0.01037	0.02055	**0.00719**	0.04700	0.03471	**0.02487**	0.08027	0.03373
	123-bus	0.00464	0.00301	0.00900	0.00555	**0.00258**	0.01294	0.00800	0.02017	0.02173	**0.00766**
	141-bus	0.00323	0.00234	0.00601	0.00608	**0.00182**	0.01070	**0.00648**	0.01501	0.02134	0.00802
	533-bus	0.00067	-	-	-	**0.00037**	0.00362	-	-	-	**0.00278**
	906-bus	0.02501	-	-	-	**0.00358**	0.02571	-	-	-	**0.00949**
3ϕ	13-bus	0.06549	-	-	-	**0.05147**	0.18626	-	-	-	**0.14138**
	37-bus	0.06294	-	-	-	**0.05907**	0.22921	-	-	-	**0.20931**
	123-bus	0.03884	-	-	-	**0.02379**	0.13387	-	-	-	**0.12065**

1ϕ stands for balanced single-phase equivalent networks, and 3ϕ stands for unbalanced three-phase networks.
The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

der three types of load scenarios: *base*, *high*, and *random* load.

1) Base Load Evaluation: We first assess the LinDistFlow approximation accuracy when using the optimized parameter values on the base loading scenarios in the test cases. Table I shows the OLDF performance metrics, notably the maximum and average voltage magnitude estimation errors ($\varepsilon_{\max}$ and ε_{avg}), compared to the alternative models PLPF, LoDF, LDF, and DLPF. As shown in this table, the OLDF model consistently outperforms its counterparts for all the test cases.

Detailing two examples, Fig. 7 shows the voltage profiles for the `37-bus` and the modified `123-bus` test cases with the base case loading. While the voltages from the traditional LinDistFlow overestimate the true values, the optimized parameters result in a close alignment with the true DistFlow solution.

2) High Load Evaluation: Following the methodology described in [19], we generated high-load scenarios by scaling the base loads with a factor that ranges from $[-2,-1]\cup[1,2]$ at a granularity of $\frac{1}{14}$, yielding 30 distinct test scenarios. As shown in Table II, our proposed OLDF algorithm surpasses the others in reducing the average voltage estimation error (ε_{avg}) across nearly all scenarios with the exception of the `22-bus` case. For this case, our OLDF results were better than all except the PLPF approximation where the average error was still quite close (0.00020 p.u. for PLPF vs. 0.00028 p.u. for OLDF). Regarding the maximum errors in the high-load scenarios, no individual approximation consistently dominated the others across all test cases. However, we note that summing the maximum errors across all test cases reveals that the OLDF parameters lead to the best performance overall for this metric. These results show that OLDF parameters trained with scenarios around base-load conditions nevertheless perform well for high-load conditions. Furthermore, as an adaptive power flow, the OLDF approximation has the ability to tailor the parameters to perform even better for these conditions by including more training scenarios associated with high loading.

3) Random Load Evaluation: The OLDF approximation's accuracy is further analyzed for random loading conditions using $10,000$ scenarios sampled from a uniform distribution of 0% to 150% of the base load. Similar to the base and high load conditions, the performance metrics for the random load conditions detailed in Table III also show the OLDF approximation's dominance, with this approximation having the smallest max and average voltage magnitude errors ($\varepsilon_{\max}$ and ε_{avg}). Across all test cases, the OLDF accuracy improvement over traditional LDF ranges from 26.42% to 91.67% for average error (ε_{avg}) and from 20.94% to 87.78% for maximum error (ε_{max}). Compared to the best of PLPF and

LoDF, OLDF's improvement ranges from 10% to 80% for average error and from 5.56% to 80.70% for max error.

E. Computational Efficiency

As shown in Fig. 2, our algorithm computes LinDistFlow parameters during an offline phase where ample computing time is available. These parameters are then used in online applications where computing time may be limited. Thus, the training process for our proposed algorithm requires computational tractability consistent with an offline context. As shown by the training times in Table IV that range from 0.6451 seconds for the `22-bus` case to 25.3917 seconds for the `123-bus` case, leveraging mature optimization methods like TNC enables acceptable scalability for the training phase.

The computation times for online uses of the optimized LinDistFlow parameters depend on the particular application for which they are employed. However, since the only changes are to the parameter values and not the mathematical form of the LinDistFlow expressions, online computation times with our optimized parameters should be comparable to existing LinDistFlow approximations. To show this, the average calculation times for the $10,000$ random load scenarios in the experiment in Section V-D3 ranged from 0.0004 to 0.1156 seconds, with differences of less than 2% between the LinDistFlow using traditional parameters and our optimized parameters.

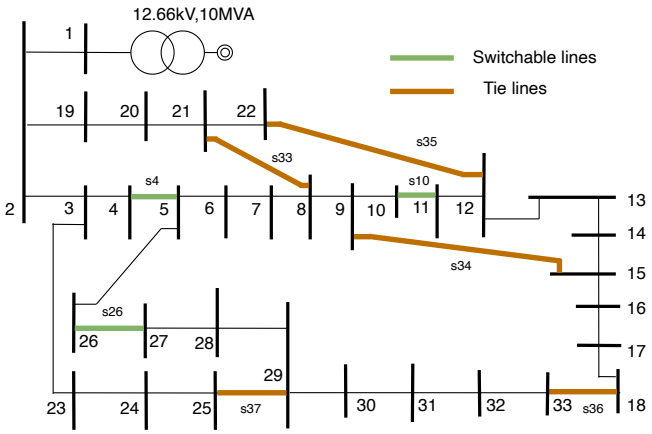


Fig. 8. The `IEEE 33-bus` system with 3 switchable lines and 5 tie-lines.

F. Topology Analysis

Engineers frequently need to both assess the impacts of topology changes for contingency assessments and optimize the topology of distribution systems for voltage management, loss minimization, outage restoration, etc. [55]. Thus, it is important to study the OLDF model's accuracy across topologies.

Addressing topology changes with the LinDistFlow approximation can be approached in various ways. For example, the traditional LinDistFlow approximation allows for straightforward topology adjustments by updating the $\mathbf{D}_r$ and $\mathbf{D}_x$ matrices with new values corresponding to the altered topology to calculate voltages. Similarly, in our optimization-based algorithm, adjustments can be made by excluding optimized parameters for removed lines and incorporating

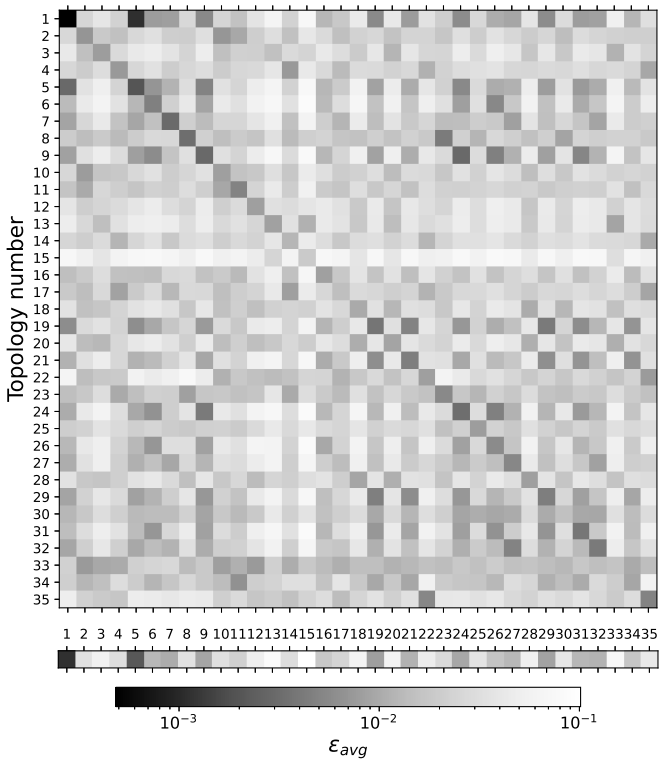


Fig. 9. Heatmap visualization of average error metric (ε_{avg}) across 35 topologies for the `IEEE 33-bus` test case, with a log-scaled color representation to highlight performance variations. The accompanying vector plot below the heatmap quantitatively assesses the LinDistFlow approximation's baseline performance across the same topologies, facilitating a direct comparison of adaptability and optimization effectiveness.

original resistance and reactance values for new lines in $\mathbf{D}_r^{opt}$ and $\mathbf{D}_x^{opt}$, without altering the bias parameters ($\boldsymbol{\gamma}^{opt}$, $\boldsymbol{\rho}^{opt}$, and $\boldsymbol{\varrho}^{opt}$). Successfully maintaining performance with this strategy indicates that our algorithm adapts well to different network topologies, avoiding overfitting to a specific configuration. Alternatively, optimizing parameters specifically for each topology through dedicated optimization could enhance accuracy at the cost of increased computational time and storage for the additional parameters needed. Nevertheless, such calculations could be efficiently executed in parallel for a specified set of topologies, making this process suitable for high-performance computing environments, as each topology's optimization process operates independently of others.

To explore these different approaches, we describe a small-scale experiment. Fig. 8 depicts the `IEEE 33-bus` distribution system, as described in [13], which has 33 nodes and 37 lines. We consider a version of this system with eight switchable lines, specifically, lines 4, 10, and 26 are normally closed switches (NCS) and lines 33 to 37 are tie lines or normally open switches (NOS). With these switchable lines, we can create 35 distinct and valid (i.e., radial and connected) topologies out of 56 possible topologies, as listed in Table V.

We next evaluate the adaptability of optimized parameters, i.e., assess how well parameters optimized for one topology perform on others. To accomplish this, we performed the

TABLE III
MODEL EVALUATION - RANDOM LOAD

	Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{OLDF}$
1ϕ	22-bus	0.00024	0.00010	0.00220	**0.00002**	0.00090	0.00057	0.00477	**0.00011**
	33-bus	0.00114	0.00051	0.00262	**0.00019**	0.00443	0.00312	0.00662	**0.00124**
	69-bus	0.00077	0.00051	0.00101	**0.00017**	0.00918	0.00816	0.00476	**0.00347**
	85-bus	0.00278	0.00056	0.00261	**0.00031**	0.00735	0.00321	0.00347	**0.00257**
	123-bus	0.00120	0.00080	0.00392	**0.00072**	0.00300	0.00227	0.00624	**0.00139**
	141-bus	0.00084	0.00035	0.00273	**0.00013**	0.00241	0.00108	0.00516	**0.00102**
	533-bus	0.00018	-	-	**0.00013**	0.00097	-	-	**0.00061**
	906-bus	0.02548	-	-	**0.00051**	0.02569	-	-	**0.00265**
3ϕ	13-bus	0.02703	-	-	**0.01335**	0.13930	-	-	**0.08872**
	37-bus	0.01438	-	-	**0.01058**	0.07639	-	-	**0.05453**
	123-bus	0.01274	-	-	**0.00637**	0.05455	-	-	**0.03506**

1ϕ stands for balanced single-phase equivalent networks, and 3ϕ stands for unbalanced three-phase networks.
The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

TABLE IV
COMPUTATION TIMES IN SECONDS

Test case	22-bus	33-bus	69-bus	85-bus	123-bus	141-bus	533-bus	906-bus	13-bus	37-bus	123-bus
t_{train}	0.6451	0.7852	4.9879	2.7182	2.3093	2.6942	3.4019	7.1058	1.2611	6.7772	25.3917
t_{base}	0.0071	0.0092	0.0161	0.0174	0.0112	0.0114	0.1025	0.3061	0.0006	0.0539	1.1495
t_{10000}	0.0004	0.0005	0.0008	0.0009	0.0006	0.0014	0.0023	0.0042	0.0182	0.0362	0.1156

t_{train}: Offline computation time to train the parameters.
t_{base}: Online computation time for the base case loading (i.e., one scenario).
t_{10000}: Online per-scenario time averaged over $10,000$ random scenarios.

TABLE V
TOPOLOGY CHANGES: OPENED AND CLOSED LINES

Opened	Closed	Opened	Closed	Opened	Closed
-	-	(4, 10)	(34, 35)	(10, 26)	(35, 36)
(4)	(33)	(4, 10)	(34, 37)	(10, 26)	(35, 37)
(4)	(35)	(4, 10)	(35, 36)	(10, 26)	(36, 37)
(4)	(37)	(4, 10)	(35, 37)	(4, 10, 26)	(33, 34, 35)
(10)	(34)	(4, 10)	(36, 37)	(4, 10, 26)	(33, 34, 36)
(10)	(35)	(4, 26)	(33, 36)	(4, 10, 26)	(33, 35, 36)
(10)	(36)	(4, 26)	(33, 37)	(4, 10, 26)	(33, 35, 37)
(26)	(36)	(4, 26)	(35, 36)	(4, 10, 26)	(33, 36, 37)
(26)	(37)	(4, 26)	(35, 37)	(4, 10, 26)	(34, 35, 36)
(4, 10)	(33, 34)	(4, 26)	(36, 37)	(4, 10, 26)	(34, 35, 37)
(4, 10)	(33, 35)	(10, 26)	(34, 35)	(4, 10, 26)	(34, 36, 37)
(4, 10)	(33, 36)	(10, 26)	(34, 36)		

These are all the valid switching combinations that lead to connected radial configurations for the `IEEE 33-bus` test case shown in Fig. 8. Note that the table indicates the changes from the configuration shown in Fig. 8.

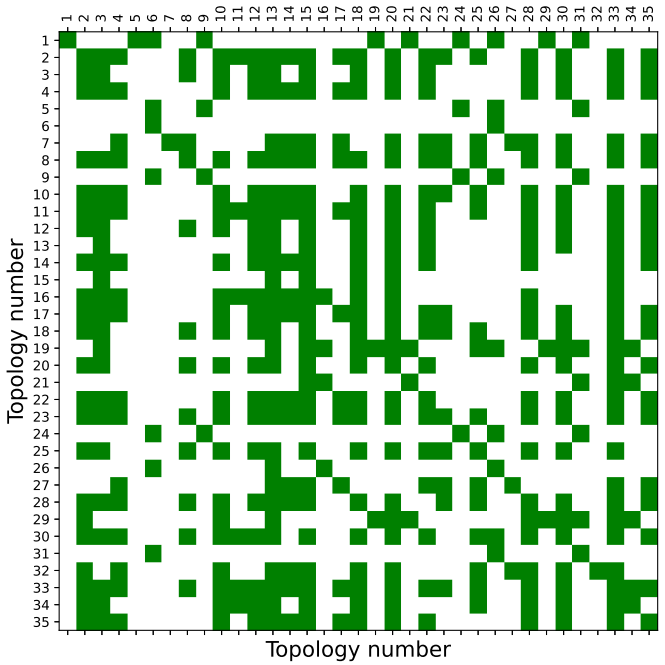


Fig. 10. Binary comparison matrix between the traditional LinDistFlow approximation's performance and optimized parameters from our proposed algorithm across 35 topologies in the `IEEE 33-bus` test case. Each column represents a topology, with green indicating that the optimized parameters for the topology associated with the corresponding row outperform the traditional LinDistFlow approximation and white indicating that the traditional LinDistFlow approximation performance is better.

offline training phase of our proposed algorithm using the same setup as before across the 35 network topologies to obtain a dataset with 35 sets of optimized coefficients and bias parameters. We tested the performance of these optimized parameters using the same $10,000$ test samples as before on each topology.

Illustrating this cross-topology assessment, Fig. 9 shows the performance of optimized parameters considering all 35 topologies. Specifically, the heatmap at the top of this figure shows the average error metric, ε_{avg}, of the optimized parameters for a given topology (rows) when applied to different topologies (columns). The horizontal vector plot at the bottom of the figure shows the performance of the traditional LinDistFlow approximation on the same 35 topologies. The heatmap employs a logarithmic scale for color representation to enable comparisons across a broad range of error magnitudes. The results underscore the variability in the algorithm's adaptability, with darker shades indicating lower errors (better performance) and lighter shades denoting higher errors. The vector plot beneath the heatmap contrasts the overall baseline performance of the traditional LinDistFlow approximation.

The matrix in Fig. 10 further illustrates the perfor-

mance of our optimized parameters relative to the traditional LinDistFlow approximation (i.e., comparing each row from heatmap in Fig. 9 to the horizontal vector plot at the bottom of the figure). This matrix employs a binary color coding—green for topologies where the optimized parameters outperform the traditional LinDistFlow approximation and white where they do not. For example, row 1 in this matrix shows that training the parameters using the base topology results in the optimized parameters outperforming the traditional LinDistFlow approximation on topologies 1, 5, 6, 9, 19, 21, 24, 26, 29, and 31. As another example, training the parameters using topology 2 leads to superior performance on topologies $\{[2,4] \cup 8 \cup [10,15] \cup [17,18] \cup 20 \cup [22,23] \cup 25 \cup 28 \cup 30 \cup 33 \cup 35\}$.

These visualizations show the nuanced performance of the optimized parameters vis-à-vis the traditional LinDistFlow approximation across topologies. Our future work aims to cluster topologies for which jointly optimized parameter values can provide accurate LinDistFlow approximations.

The adaptability of the OLDF framework also extends to modeling active control devices like switched capacitors and voltage regulators. In scenarios where these devices are in a fixed configuration or are locally controlled, the offline training process has an advantage over traditional methods by implicitly learning the average impact of their behavior from the power flow solution samples. For optimization problems where device settings are decision variables, the OLDF model is also well-suited. The reactive power injections from capacitor banks can be treated as direct inputs (q_n) to the model. Furthermore, different tap positions of a voltage regulator can be modeled as distinct network configurations, allowing for tailored OLDF parameters to be trained offline for each setting, leveraging the same parallelizable approach used for topology changes. This flexibility allows the OLDF model to create more accurate, system-specific linearizations for a wider range of practical operating scenarios.

G. Illustrative Application: Hosting Capacity Computation

As an illustrative example application, this section demonstrates the use of the proposed optimized LinDistFlow model to determine the hosting capacity of inverter-based generation units. Following [56], the hosting capacity problem is:

$$\underset{p_n^g, q_n^g}{\text{minimize}} \quad \sum_{n \in \hat{\mathcal{N}}} \left(\frac{(\overline{p_n^g} - p_n^g)^2}{\overline{p_n^g}} + \xi \frac{{q_n^g}^2}{\overline{s_n^g}} \right) \tag{15a}$$

$$\text{s.t.} \quad 0 \le p_n^g \le \overline{p_n^g}, \qquad |q_n^g| \le \sqrt{\overline{s_n^g}^2 - {p_n^g}^2}, \ \forall n \in \hat{\mathcal{N}} \tag{15b}$$

$$v_0 = 1, \quad \underline{v_n} \le v_n \le \overline{v_n}, \qquad \forall n \in \mathcal{N}' \tag{15c}$$

$$\sqrt{P_n^2 + Q_n^2} \le S_n, \qquad \forall n : (\pi_n, n) \in \mathcal{E} \tag{15d}$$

$$\sqrt{P_T^2 + Q_T^2} \le S_T, \tag{15e}$$

$$P_T = \sum_{n:(0,n)\in\mathcal{E}} P_{0n}, \quad Q_T = \sum_{n:(0,n)\in\mathcal{E}} Q_{0n}, \tag{15f}$$

$$\text{LDF (3) or OLDF (4)}, \tag{15g}$$

where $\hat{\mathcal{N}}$ is the set of buses with inverter-based generators, p_n^g and q_n^g are the active and reactive power generation, $\overline{p_n^g}$ and $\overline{s_n^g}$ are their maximum capacities, S_n and S_T are the upper bounds on the line and transformer flows, and ξ is a weighting factor controlling the tradeoff between active power and reactive power utilization. The constraints ensure adherence to power generation limits (15b) and voltage regulation requirements (15c) along with line (15d) and transformer capacities (15e)–(15f).

An evaluation using the `IEEE 33-bus` test system illustrates the OLDF model's effectiveness. We set $\overline{s_n^g} = 0.6$ MVA with a 0.98 power factor, $\xi = 0.02$, and voltage limits between 1.05 and 0.95 per unit, with a substation capacity of 10 MVA.

Upon solving (15) with both LDF and our proposed OLDF models, we obtain the optimal active and reactive power settings for the inverter-based generation units. By assessing these optimal settings using the original nonlinear DistFlow model, we compare the performance of the OLDF and LDF approximations. To clarify the comparison methodology, we first solved the optimization problem (15) using the LDF model to find its proposed optimal generator setpoints. We then solved the same problem using the OLDF model to find a different set of optimal setpoints. Finally, to determine the true feasibility and performance of each solution, we evaluated *both* sets of setpoints using the accurate nonlinear DistFlow model. Fig. 11 plots the actual voltage profiles resulting from this validation. As shown in this figure, while the traditional LinDistFlow approximation leads to voltage violations at certain buses within the hosting capacity problem (15), the application of the proposed OLDF model avoids such violations. The voltage violations resulting from the LDF-based optimization in Fig. 11 occur because traditional LDF inaccurately models the system's voltage response to the high generation injections inherent in hosting capacity analysis. This can lead LDF to identify generation setpoints that appear feasible within its simplified model but are, in fact, infeasible in the true nonlinear system. Conversely, OLDF's parameters, having been optimized for broad accuracy across diverse conditions, provide a more accurate system representation, thus yielding genuinely feasible solutions. Such discrepancies, where LDF-based decisions prove unreliable when operating near system limits, are generally anticipated given OLDF's consistently superior performance demonstrated across various test cases and loading scenarios (Tables I, II, and III).

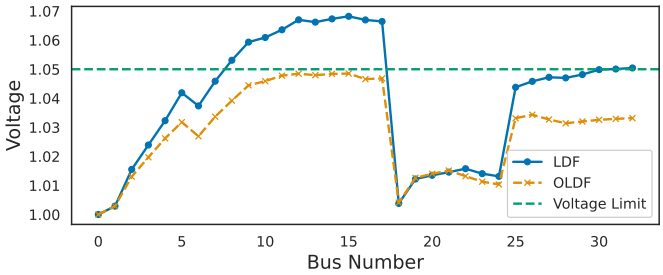


Fig. 11. Voltage profile of the `IEEE 33-bus` test case after feeding the active/reactive power injections obtained from solving (15) with LDF (blue curve) and OLDF (orange curve) into the DistFlow equations.

VI. Conclusion

The LinDistFlow approximation is often used to improve the computational tractability of optimization problems in distribution systems. This paper introduces a new algorithm that enhances the accuracy of the LinDistFlow approximation for single- and three-phase network models. Inspired by machine learning methods, the algorithm's offline phase optimizes the approximation's coefficients and biases using analytically derived sensitivities within the TNC optimization method. These optimized parameters then provide increased accuracy when used in various applications. Numerical tests demonstrate the algorithm's effectiveness, showing better alignment with nonlinear DistFlow solutions compared to the traditional LinDistFlow and other recent alternatives. Application to a hosting capacity problem further highlights its advantages.

Since our optimized formulation has the same linear form as the traditional LinDistFlow approximation and performs well for varying topologies, the accuracy advantages of the proposed approach can be directly exploited in a wide range of other applications such as those in [11]–[13], [15], [16], [21]–[25]. Our future work aims to explore such applications. Building on prior literature such as [57]–[59], our future work will also focus on extending the OLDF framework to more effectively optimize setpoints for active control devices, such as voltage regulators and switchable capacitor banks. While the current OLDF enhances the modeling of the passive network (with capacitor bank actions represented by their reactive power injections, q_n, and fixed regulator effects implicitly learned during training), its optimal integration with dynamically operating controls warrants dedicated investigation. Key research directions include tailoring OLDF training strategies to capture the diverse system states induced by these controls and developing methods for adapting OLDF parameters based on their operational status, potentially drawing insights from our topology adaptation analysis. Such advancements aim to further enhance OLDF's practical utility in actively controlled distribution systems.

References

[1] B. Stott, "Review of load-flow calculation methods," *Proc. IEEE*, vol. 62, no. 7, pp. 916–929, 1974.

[2] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-hard," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 798–801, January 2016.

[3] T. J. Overbye, X. Cheng, and Y. Sun, "A comparison of the AC and DC power flow models for LMP calculations," in *37th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2004.

[4] S. Misra, L. A. Roald, M. Vuffray, and M. Chertkov, "Fast and robust determination of power system emergency control actions," in *10th IREP Symp. Bulk Power Syst. Dynamics Control*, August 2017.

[5] L. A. Roald, D. Pozo, A. Papavasiliou, D. K. Molzahn, J. Kazempour, and A. Conejo, "Power Systems Optimization under Uncertainty: A Review of Methods and Applications," *Electric Power Syst. Res.*, vol. 214, no. 108725, January 2023, presented at the *22nd Power Syst. Comput. Conf. (PSCC)*.

[6] C. Barrows, S. Blumsack, and P. Hines, "Correcting optimal transmission switching for AC power flows," in *47th Hawaii Int. Conf. Syst. Sci. (HICSS)*, January 2014, pp. 2374–2379.

[7] A. Castillo, C. Laird, C. A. Silva-Monroy, J.-P. Watson, and R. P. O'Neill, "The unit commitment problem with AC optimal power flow constraints," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4853–4866, 2016.

[8] B. Austgen, E. Kutanoglu, J. J. Hasenbein, and S. Santoso, "Comparisons of two-stage models for flood mitigation of electrical substations," *INFORMS J. Comput.*, vol. 37, no. 2, pp. 270–292, 2025.

[9] N. Rhodes, E. Haag, and L. Roald, "Long solution times or low solution quality: On trade-offs in choosing a power flow formulation for the optimal power shut-off problem," *Electric Power Syst. Res.*, vol. 234, no. 110713, 2024, presented at the *23rd Power Syst. Comput. Conf. (PSCC)*.

[10] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Found. Trends Electr. Energy Syst.*, vol. 4, no. 1-2, pp. 1–221, 2019.

[11] M. Baran and F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 725–734, Jan 1989.

[12] ——, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 735–743, Jan 1989.

[13] ——, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, April 1989.

[14] L. Gan and S. H. Low, "Convex relaxations and linear approximation for optimal power flow in multiphase radial networks," in *18th Power Syst. Comput. Conf. (PSCC)*, August 2014.

[15] V. Kekatos, L. Zhang, G. B. Giannakis, and R. Baldick, "Voltage regulation algorithms for multiphase power distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3913–3923, September 2016.

[16] B. A. Robbins and A. D. Domínguez-García, "Optimal reactive power dispatch for voltage regulation in unbalanced distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2903–2913, July 2016.

[17] Z. Yang, H. Zhong, A. Bose, T. Zheng, Q. Xia, and C. Kang, "A linearized OPF model with reactive power and voltage magnitude: A pathway to improve the MW-only DC OPF," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1734–1745, 2017.

[18] E. Schweitzer, S. Saha, A. Scaglione, N. G. Johnson, and D. Arnold, "Lossy DistFlow formulation for single and multiphase radial feeders," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1758–1768, May 2020.

[19] M. Marković and B.-M. Hodge, "Parameterized linear power flow for high fidelity voltage solutions in distribution systems," *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4391–4403, September 2023.

[20] V. Madani, M. Karrari, and A. Deihimi, "On the application of the branch DistFlow using second-order conic programming in microgrids and active distribution systems," *Electric Power Syst. Res.*, vol. 242, p. 111226, March 2025.

[21] K. Baker, A. Bernstein, E. Dall'Anese, and C. Zhao, "Network-cognizant voltage droop control for distribution grids," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2098–2108, 2017.

[22] D. B. Arnold, M. Sankur, R. Dobbe, K. Brady, D. S. Callaway, and A. Von Meier, "Optimal dispatch of reactive power for voltage regulation and balancing in unbalanced distribution systems," in *IEEE Power Energy Soc. Gen. Meeting*, 2016.

[23] R. Mieth and Y. Dvorkin, "Data-driven distributionally robust optimal power flow for distribution systems," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 363–368, 2018.

[24] R. K. Gupta and D. K. Molzahn, "Improving fairness in photovoltaic curtailments via daily topology reconfiguration for voltage control in power distribution networks," *arXiv:2403.07853*, 2024.

[25] S. Bose, K. Chen, and Y. Zhang, "On LinDistFlow model congestion pricing: Bounding the changes in power tariffs," in *32nd IEEE Int. Symp. Ind. Electron. (ISIE)*, 2023.

[26] X. Chen, W. Wu, and B. Zhang, "Robust capacity assessment of distributed generation in unbalanced distribution networks incorporating anm techniques," *IEEE Trans. Sustain. Energy*, vol. 9, no. 2, pp. 651–663, April 2018.

[27] Y. Jiang, X. Dai, F. Zahn, and V. Hagenmeyer, "Enhanced flexibility aggregation using LinDistFlow model with loss compensation," *arXiv:2505.01715*, 2025.

[28] S. Misra, D. K. Molzahn, and K. Dvijotham, "Optimal adaptive linearizations of the AC power flow equations," in *20th Power Syst. Comput. Conf. (PSCC)*, June 2018.

[29] M. Hohmann, J. Warrington, and J. Lygeros, "Optimal linearizations of power systems with uncertain supply and demand," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1504–1512, March 2019.

[30] T. Mühlpfordt, V. Hagenmeyer, D. K. Molzahn, and S. Misra, "Optimal adaptive power flow linearizations: Expected error minimization using polynomial chaos expansion," in *IEEE Milan PowerTech*, 2019.

[31] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, "Data-driven power flow linearization: A regression approach," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2569–2580, May 2019.

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

[33] J. Chen and L. A. Roald, "Topology-adaptive piecewise linearization for three-phase power flow calculations in distribution grids," in *55th North American Power Symp. (NAPS)*, 2023.

[34] P. Buason, S. Misra, and D. K. Molzahn, "A sample-based approach for computing conservative linear power flow approximations," *Electric Power Syst. Res.*, vol. 212, p. 108579, 2022, presented at the *22nd Power Syst. Comput. Conf. (PSCC)*.

[35] B. Taheri and D. K. Molzahn, "Optimizing parameters of the DC power flow," *Electric Power Syst. Res.*, vol. 235, no. 110719, Oct. 2024, presented at the *23rd Power Syst. Comput. Conf. (PSCC)*.

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

[37] M. Jia, G. Hug, N. Zhang, Z. Wang, and Y. Wang, "Tutorial on data-driven power flow linearization–Part I: Challenges and training algorithms," preprint available at https://doi.org/10.3929/ethz-b-000606654, 2023.

[38] ——, "Tutorial on data-driven power flow linearization–Part II: Supportive techniques and experiments," preprint available at https://doi.org/10.3929/ethz-b-000606656, 2023.

[39] B. Taheri and D. K. Molzahn, "Improving the accuracy of DC Optimal Power Flow formulations via parameter optimization," *arXiv:2410.11725*, October 2024.

[40] A. Rosemberg and M. Klamkin, "Differentiable optimization for deep learning-enhanced DC approximation of AC optimal power flow," *arXiv:2504.01970*, March 2025.

[41] Y. Chen and M. K. Singh, "Optimally linearizing power flow equations for improved power system dispatch," *arXiv:2504.03076*, April 2025.

[42] B. Taheri and D. K. Molzahn, "AC-informed DC optimal transmission switching problems via parameter optimization," to appear in *IEEE Trans. Power Syst.*, 2025.

[43] P. Buason, S. Misra, J. P. Watson, and D. K. Molzahn, "Adaptive power flow approximations with second-order sensitivity insights," *IEEE Trans. Power Syst.*, vol. 40, no. 3, pp. 2648–2660, May 2025.

[44] Y.-h. Cho and H. Zhu, "Data-driven modeling of linearizable power flow for large-scale grid topology optimization," *arXiv:2409.13956*, 2024.

[45] P. Buason, S. Misra, and D. K. Molzahn, "Sample-based piecewise linear power flow approximations using second-order sensitivities," in *IEEE Kiel PowerTech*, June 2025.

[46] W. Li, Y. Zhao, C. Wang, J. Liu, Y. Wu, and D. Liu, "Data-driven distributionally robust optimal power flow for distribution grids under Wasserstein ambiguity sets," *Electronics*, vol. 14, no. 4, p. 822, February 2025.

[47] S. H. Low, "Convex relaxation of optimal power flow–Part I: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 15–27, 2014.

[48] S. Bose, S. H. Low, T. Teeraratkul, and B. Hassibi, "Equivalent relaxations of optimal power flow," *IEEE Trans. Automat. Control*, vol. 60, no. 3, pp. 729–742, 2015.

[49] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.

[50] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer Science & Business Media, 2006.

[51] S. G. Nash, "Newton-type minimization via the Lanczos method," *SIAM J. Numerical Analy.*, vol. 21, no. 4, pp. 770–788, 1984.

[52] S. Bolognani and S. Zampieri, "On the existence and linear approximation of the power flow solution in power distribution networks," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 163–172, 2015.

[53] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[54] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *20th Power Syst. Comput. Conf. (PSCC)*, 2018.

[55] M. Mahdavi, H. H. Alhelou, N. D. Hatziargyriou, and F. Jurado, "Reconfiguration of electric power distribution systems: Comprehensive review and classification," *IEEE Access*, vol. 9, pp. 118 502–118 527, 2021.

[56] S. Zhan, J. Morren, W. van den Akker, A. van der Molen, N. G. Paterakis, and J. Slootweg, "Fairness-incorporated online feedback optimization for real-time distribution grid management," *IEEE Trans. Smart Grid*, vol. 15, no. 2, pp. 1792–1806, 2024.

[57] W. Wu, Z. Tian, and B. Zhang, "An exact linearization method for OLTC of transformer in branch flow model," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2475–2476, May 2017.

[58] M. Bazrafshan, N. Gatsis, and H. Zhu, "Optimal tap selection of step-voltage regulators in multi-phase distribution networks," in *20th Power Syst. Comput. Conf. (PSCC)*, June 2018.

[59] ——, "Optimal power flow with step-voltage regulators in multi-phase distribution networks," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4228–4239, 2019.

APPENDIX
DERIVATION OF THREE-PHASE LINDISTFLOW (LINDIST3FLOW) MODEL

This appendix presents a detailed derivation of the linearized three-phase power flow model, LinDist3Flow [14], [22]. Specifically, this appendix derives the voltage update equation and the coefficient matrices $\mathbb{H}^P$ and $\mathbb{H}^Q$ that are presented in equations (8), (9), and (10) in Section IV of the main paper. Consider two adjacent buses in the distribution feeder, denoted as $(n,k) \in \mathcal{N}$. We first express the power flow equations in vector form, as follows:

$$\mathtt{v}_n = \mathtt{v}_k + \mathtt{Z}_k \mathtt{I}_k, \tag{16}$$

$$\mathtt{I}_n = \mathtt{i}_n + \sum_{k:n\to k} \mathtt{I}_k, \tag{17}$$

where $\mathtt{v}_n = \begin{bmatrix} V^a & V^b & V^c \end{bmatrix}^\top_n$ is the vector of voltage phasors at node n, $\mathtt{i}_n = \begin{bmatrix} i^a & i^b & i^c \end{bmatrix}^\top_n$ is the vector of load currents at node n, $\mathtt{I}_n = \begin{bmatrix} I^a & I^b & I^c \end{bmatrix}^\top_n$ is the vector of current phasors entering node n. Next, we multiply both sides of (16) by their complex conjugates and similarly multiply both sides of (17) by $\mathtt{v}_n$. This gives us the following equations:

$$\begin{aligned} \mathtt{v}_n\mathtt{v}_n^* &= \mathtt{v}_k\mathtt{v}_k^* + \mathtt{Z}_k\mathtt{I}_k\mathtt{v}_k^* + \mathtt{v}_k\mathtt{I}_k^*\mathtt{Z}_k^* + \mathtt{Z}_k\mathtt{I}_k\mathtt{I}_k^*\mathtt{Z}_k^*, \\ &= \mathtt{v}_k\mathtt{v}_k^* + 2\Re\{\mathtt{v}_k\mathtt{I}_k^*\mathtt{Z}_k^*\} + \mathtt{Z}_k\mathtt{I}_k\mathtt{I}_k^*\mathtt{Z}_k^*, \end{aligned} \tag{18}$$

$$\mathtt{v}_n\mathtt{I}_n^* = \mathtt{v}_n\mathtt{i}_n^* + \sum_{k:n\to k} (\mathtt{v}_k + \mathtt{Z}_k\mathtt{I}_k)\mathtt{I}_k^*. \tag{19}$$

Similar to the derivation of the LinDistFlow model, we can ignore the loss terms in (18) and (19), which results in:

$$\mathtt{v}_n\mathtt{v}_n^* \approx \mathtt{v}_k\mathtt{v}_k^* + 2\Re\{\mathtt{v}_k\mathtt{I}_k^*\mathtt{Z}_k\}, \tag{20}$$

$$\mathtt{v}_n\mathtt{I}_n^* \approx \mathtt{v}_n\mathtt{i}_n^* + \sum_{k:n\to k} \mathtt{v}_k\mathtt{I}_k^*. \tag{21}$$

Equations (20) and (21) represent 3×3 matrix equations. To focus on (21), we apply the power equation $S_{\phi,k} = V_{\phi,k}I^*_{\phi,k}$ and collect the diagonal terms into a vector equation where $S_{\phi,k}$ is the complex power phasor entering node k at phase ϕ:

$$\mathtt{S}_n \approx \mathtt{s}_n + \sum_{k:n\to k} \mathtt{S}_k, \tag{22}$$

where $\mathtt{s}_n$ is the vector of complex loads at node n. Now, returning to (20), we can expand $\mathtt{I}_k$ using the power equation $S_{\phi,k} = V_{\phi,k}I^*_{\phi,k}$, which yields:

$$\mathtt{v}_n\mathtt{v}_n^* \approx \mathtt{v}_k\mathtt{v}_k^* + 2\Re\left\{\mathtt{v}_k\left[S_a v_a^{-1} S_b v_b^{-1} S_c v_c^{-1}\right]_{nk}\mathtt{Z}_k^*\right\}, \tag{23}$$

$$\mathtt{v}_n\mathtt{v}_n^* \approx \mathtt{v}_k\mathtt{v}_k^*$$

$$+2\Re\left\{\begin{bmatrix} S_a & V_aS_bV_b^{-1} & V_aS_cV_c^{-1} \\ V_bS_aV_a^{-1} & S_b & V_bS_cV_c^{-1} \\ V_cS_aV_a^{-1} & V_acS_bV_b^{-1} & S_c \end{bmatrix} \mathsf{Z}_{nk}^*\right\}_k. \tag{24}$$

We now assume the ratios of voltage phasors are constant:

$$V_{a,k}V_{b,k}^{-1} \approx \alpha, \quad V_{b,k}V_{c,k}^{-1} \approx \alpha, \quad V_{a,k}V_{c,k}^{-1} \approx \alpha^2. \tag{25}$$

Here, the constants α and α^2 are defined as:

$$\alpha = 1\angle 120^\circ = -\frac{1}{2} + \mathrm{j}\frac{\sqrt{3}}{2}, \quad \alpha^2 = 1\angle 240^\circ = -\frac{1}{2} - \mathrm{j}\frac{\sqrt{3}}{2}. \tag{26}$$

Substituting these values from (25) and (26) into (24), we obtain the final form:

$$\mathsf{v}_n\mathsf{v}_n^* \approx \mathsf{v}_k\mathsf{v}_k^* + 2\Re\left\{\begin{bmatrix} S_a & \alpha S_b & \alpha^2 S_c \\ \alpha^2 S_a & S_b & \alpha S_c \\ \alpha S_a & \alpha^2 S_b & S_c \end{bmatrix}_k \mathsf{Z}_{nk}^*\right\}. \tag{27}$$

We are mainly concerned with the diagonal elements of (27), which we gather into a 3×1 vector:

$$\mathbb{V}_n \approx \mathbb{V}_k + 2\Re\left\{\begin{bmatrix} Z_{aa,nk}^* S_{a,k} + \alpha Z_{ab,nk}^* S_{b,k} + \alpha^2 Z_{ac,nk}^* S_{c,k} \\ \alpha^2 Z_{ba,nk}^* S_{a,k} + Z_{bb,nk}^* S_{b,k} + \alpha Z_{bc,nk}^* S_{c,k} \\ \alpha Z_{ca,nk}^* S_{a,k} + \alpha^2 Z_{cb,nk}^* S_{b,k} + Z_{cc,nk}^* S_{c,k} \end{bmatrix}\right\}, \tag{28}$$

$$\mathbb{V}_n \approx \mathbb{V}_k + 2\Re\left\{\begin{bmatrix} Z_{aa}^* + \alpha Z_{ab}^* + \alpha^2 Z_{ac}^* \\ \alpha^2 Z_{ba}^* + Z_{bb}^* + \alpha Z_{bc}^* \\ \alpha Z_{ca}^* + \alpha^2 Z_{cb}^* + Z_{cc}^* \end{bmatrix}_{nk} \begin{bmatrix} S_a \\ S_b \\ S_c \end{bmatrix}_k\right\}, \tag{29}$$

where $\mathbb{V}_n = \begin{bmatrix} v^a & v^b & v^c \end{bmatrix}_n^\top$ is the vector of squared of voltage magnitudes for each phase (a, b, c) at node n. Expanding the impedance matrix entries $Z_{\phi\psi,k} = r_{\phi\psi,k} + \mathrm{j}x_{\phi\psi,k}$, and using $S_{\phi,k} = P_{\phi,k} + \mathrm{j}Q_{\phi,k}$, we simplify (28) to:

$$\mathbb{V}_n \approx \mathbb{V}_k - \mathbb{H}_{nk}^P \mathbb{P}_k - \mathbb{H}_{nk}^Q \mathbb{Q}_k, \tag{30}$$

where the matrices $\mathbb{H}_{nk}^P$ and $\mathbb{H}_{nk}^Q$ are defined as:

$$\mathbb{H}_{nk}^P = \begin{bmatrix} -2r_{aa} & r_{ab} - \sqrt{3}x_{ab} & r_{ac} + \sqrt{3}x_{ac} \\ r_{ba} + \sqrt{3}x_{ba} & -2r_{bb} & r_{bc} - \sqrt{3}x_{bc} \\ r_{ca} - \sqrt{3}x_{ca} & r_{cb} + \sqrt{3}x_{cb} & -2r_{cc} \end{bmatrix}, \tag{31}$$

$$\mathbb{H}_{nk}^Q = \begin{bmatrix} -2x_{aa} & x_{ab} + \sqrt{3}r_{ab} & x_{ac} - \sqrt{3}r_{ac} \\ x_{ba} - \sqrt{3}r_{ba} & -2x_{bb} & x_{bc} + \sqrt{3}r_{bc} \\ x_{ca} + \sqrt{3}r_{ca} & x_{cb} - \sqrt{3}r_{cb} & -2x_{cc} \end{bmatrix}. \tag{32}$$

Finally, we restate (22) for clarity:

$$S_n \approx s_n + \sum_{k:n\to k} S_k. \tag{33}$$

Equations (30)–(33) represent a linearized model for unbalanced three-phase power flow, which maps the active and reactive power injections at node k into squared voltage magnitude differences. Contributions from all phases influence each other's squared voltage magnitudes, and reducing this system to a single-phase network recovers the classical LinDistFlow model, as noted in [12].

Now we can write the (30) in matrix form as follows:

$$\mathbb{V} = \mathbb{V}_0\mathbf{1} + \mathbb{A}_3^{-1}\mathrm{bdiag}(\mathbb{H}^P)\mathbb{A}_3^{-\top}\mathbb{P} + \mathbb{A}_3^{-1}\mathrm{bdiag}(\mathbb{H}^Q)\mathbb{A}_3^{-\top}\mathbb{Q}, \tag{34}$$

where $\mathbb{V} = \begin{bmatrix} v_1^a & v_1^b & v_1^c & \cdots & v_n^a & v_n^b & v_n^c \end{bmatrix}^\top$ is the vector of squared of voltage magnitudes for each phase (a, b, c), $\mathbb{A}_3$ is the network incidence matrix, $\mathrm{bdiag}(\cdot)$ is the block diagonal operator, and $\mathbb{P} = \begin{bmatrix} p_1^a & p_1^b & p_1^c & \cdots & p_n^a & p_n^b & p_n^c \end{bmatrix}^\top$ and $\mathbb{Q} = \begin{bmatrix} q_1^a & q_1^b & q_1^c & \cdots & q_n^a & q_n^b & q_n^c \end{bmatrix}^\top$ are the active and reactive power demand vectors.

Adding parameters $\boldsymbol{\rho}_3$, $\boldsymbol{\varrho}_3$, and $\boldsymbol{\gamma}_3$ to (34) yields:

$$\mathbb{V} = \mathbb{V}_0\mathbf{1} + \mathbb{A}_3^{-1}\mathrm{bdiag}(\mathbb{H}^P)\mathbb{A}_3^{-\top}(\mathbb{P} + \boldsymbol{\rho}_3) + \\ + \mathbb{A}_3^{-1}\mathrm{bdiag}(\mathbb{H}^Q)\mathbb{A}_3^{-\top}(\mathbb{Q} + \boldsymbol{\varrho}_3) + \boldsymbol{\gamma}_3. \tag{35}$$

Babak Taheri received the Ph.D. and M.Sc. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2024. He also received the M.Sc. degree in electrical engineering from Sharif University of Technology in 2019 and the B.Sc. degree from the University of Tabriz in 2017. He is currently a Research Scientist at Hitachi Energy Research. His research interests include power systems, optimization, and machine learning.

Rahul K. Gupta (Member, IEEE) received the B.Tech. degree in electrical engineering from the National Institute of Technology (NIT), Rourkela, India, in 2014, and the M.Sc. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2018 and 2023, respectively. From October 2023 to December 2024, he was a Postdoctoral Scholar with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, supported by a grant from the Swiss National Science Foundation (SNSF). Since 2025, he has been an Assistant Professor with the School of Electrical Engineering and Computer Science (EECS), Washington State University (WSU), Pullman, WA, USA. He was the recipient of the ABB Research Award 2025 and EPFL Ph.D. Thesis Distinction in Electrical Engineering for his doctoral research.

Daniel K. Molzahn (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering and the master's of Public Affairs degree from the University of Wisconsin–Madison, Madison, WI, USA. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, and also holds an appointment as a Computational Engineer in the Energy Systems Division at Argonne National Laboratory. He was a Dow Postdoctoral Fellow in Sustainability at the University of Michigan, Ann Arbor, MI, USA, and a National Science Foundation Graduate Research Fellow at the University of Wisconsin–Madison. He was the recipient of the IEEE Power and Energy Society's Outstanding Young Engineer Award in 2021, the NSF CAREER Award in 2022, and Georgia Tech's Class of 1940 W. Roane Beard Outstanding Teacher Award in 2024.