

# Improving the Accuracy of DC Optimal Power Flow Formulations via Parameter Optimization

Babak Taheri and Daniel K. Molzahn

**Abstract**—DC Optimal Power Flow (DC-OPF) problems optimize the generators’ active power setpoints while satisfying constraints based on the DC power flow linearization. The computational tractability advantages of DC-OPF problems come at the expense of inaccuracies relative to AC Optimal Power Flow (AC-OPF) problems that accurately model the nonlinear steady-state behavior of power grids. This paper proposes an algorithm that significantly improves the accuracy of the generators’ active power setpoints from DC-OPF problems with respect to the corresponding AC-OPF problems over a specified range of operating conditions. Using sensitivity information in a machine learning-inspired methodology, this algorithm tunes coefficient and bias parameters in the DC power flow approximation to improve the accuracy of the resulting DC-OPF solutions. Employing the Truncated Newton Conjugate-Gradient (TNC) method—a Quasi-Newton optimization technique—this parameter tuning occurs during an offline training phase, with the resulting parameters then used in online computations. Numerical results underscore the algorithm’s efficacy with accuracy improvements in squared two-norm and  $\infty$ -norm losses of up to 90% and 79%, respectively, relative to traditional DC-OPF formulations.

**Index Terms**—DC optimal power flow (DC-OPF), AC-OPF, machine learning, parameter optimization.

## I. INTRODUCTION

Optimal Power Flow (OPF) is a fundamental tool for power system design and operation. OPF problems optimize system performance (e.g., minimizing generation costs or reducing voltage deviations) while satisfying both equality constraints from a power flow model and inequality constraints from operational limits. Along with economic implications [1], OPF problems are central to many algorithms for addressing uncertainty [2], expansion planning [3], improving stability margins [4], etc.

The AC power flow equations accurately model the steady-state behavior of power systems by relating the complex power injections with the voltage phasors. This motivates solution of AC Optimal Power Flow (AC-OPF) problems that incorporate the AC power flow equations. However, AC-OPF problems are computationally challenging, with these problems being non-convex and NP-hard in the worst-case [5]–[7]. These challenges are exacerbated for extensions of OPF problems that consider discrete characteristics, such as those related to transmission switching [8], [9] (where inaccuracies in DC model parameters can significantly impact the quality of switching decisions [10]) and unit commitment [11], as well as uncertainties due to variable renewable energy generation and flexible loads [2].

B. Taheri and D.K. Molzahn are with the School of Electrical and Computer Engineering, Georgia Institute of Technology. {btaheri6, molzahn}@gatech.edu. Support from NSF award #2145564.

OPF algorithms have been a subject of research for many decades [12], [13]. Most existing algorithms seek a local optimum due to the non-convex nature of OPF problems. Machine learning (ML) tools can address OPF challenges, offering computational speedups but often lacking interpretability<sup>1</sup> and consistency with physical intuition [14].

Due to these challenges, engineers often employ power flow approximations to obtain more tractable convex OPF formulations [15]. Using a common linear approximation of the AC power flow equations known as the DC power flow [16] yields the DC Optimal Power Flow (DC-OPF) problem, which is widely used for both short- and long-term planning purposes. While other advanced linearizations of the AC power flow equations exist [15], [17], our focus in this paper is on improving the accuracy of the *DC-OPF formulation* due to its widespread usage.

The DC power flow is derived under a set of standard assumptions: a flat voltage profile (magnitudes near 1.0 p.u.), small voltage angle differences, and negligible network losses and reactive power flows. These assumptions render the model computationally tractable and are often reasonably accurate for high-voltage transmission systems. However, approximation errors are introduced when these assumptions do not hold. While heuristics exist to mitigate some inaccuracies, such as embedding estimated power losses within the load data, such approaches may not be robust across diverse operating conditions [16]. Inaccuracies from the DC power flow assumptions are particularly pronounced when reactive power flows and voltage variations play a significant role, such as in heavily loaded systems [18] or systems with high penetrations of renewable generators [19]. In such cases, the DC-OPF problem may not adequately capture the complex behavior of the power system, leading to suboptimal or infeasible decisions in operational and planning contexts [8], [20]–[22].

While the standard DC-OPF formulation is widely used [16], its accuracy limitations have spurred the development of various alternatives [15]. These can be broadly categorized into those that propose new model structures, those that tune parameters of existing models, and those that are purely based on machine learning techniques. A significant body of work focuses on creating more accurate linear approximations of the AC power flow equations, e.g., [23]. The survey in [15] catalogs a wide array of these relaxations and approximations. While potentially more accurate, these models typically introduce new mathematical structures that pose significant practical implementation challenges, as the

<sup>1</sup>In this context, interpretability refers to the ability to directly relate a model’s parameters and variables to the physical principles of the power system, a feature often lacking in more complex, black-box approaches.

conventional DC-OPF structure is deeply embedded in numerous industry-grade and academic tools for complex tasks like unit commitment and transmission planning. Altering these problems' fundamental structure would require a significant overhaul of these established tools. Another research direction, more aligned with our philosophy, focuses on enhancing existing models by tuning their coefficients and bias parameters. This includes optimizing the linearization point to account for supply and demand uncertainty [24]. This paradigm of enhancing classical, interpretable models has also proven effective for other approximations, such as the LinDistFlow model for distribution systems [25]. Our prior work used a single-level optimization to tune DC power flow parameters to better match AC power flow results [26]. While effective at improving the approximation's accuracy, it did not directly optimize for the end-goal of improving the OPF solution itself. More recently, black-box machine learning models have been developed to predict AC-OPF solutions with high fidelity [27], though often at the cost of interpretability.

Our work, in contrast, occupies a middle ground. We leverage techniques from the machine learning domain, specifically differentiable optimization layers [28], [29], but our goal is to enhance a classical, interpretable model. This paper distinguishes itself by intentionally preserving the standard DC-OPF formulation to ensure immediate compatibility with existing tools. Furthermore, we introduce a bilevel optimization framework that represents a significant conceptual advance. Rather than simply matching intermediate power flow values, our method tunes the DC-OPF parameters to directly minimize the error in the final optimal generator setpoints. This solution-focused approach carves a third path between pure machine learning and classical, fixed-parameter models, enhancing accuracy while preserving the interpretability and compatibility of the standard DC-OPF model.

The DC power flow approximation is foundational to the DC-OPF problem. This approximation relates active power flow between buses  $i$  and  $j$ , denoted  $p_{ij}$ , to the phase angle difference  $\theta_i - \theta_j$  through a proportionality coefficient  $b_{ij}$ :  $p_{ij} = b_{ij}(\theta_i - \theta_j)$ . While typically derived from line parameters such as resistance  $r_{ij}$  and reactance  $x_{ij}$  via various heuristics, tailoring these parameter values to specific system and operating range characteristics can yield significant accuracy improvements. Common heuristics choose  $b_{ij} = 1/x_{ij}$  or  $b_{ij} = -\Im(1/(r_{ij} + jx_{ij}))$ , where  $\Im(\cdot)$  extracts the imaginary part of a complex argument. When resistance is nonzero ( $r_{ij} \neq 0$ ), these choices for  $b_{ij}$  produce slightly different results, impacting the accuracy of the DC power flow approximation. Selecting the optimal  $b_{ij}$  for a given application is not straightforward and depends on system characteristics and objectives [16]. Additionally, the introduction of bias parameters allows further flexibility by adjusting power injections and flows to account for shunts, HVDC infeeds, phase shifts, and line losses [16]. Two variants of the DC power flow—cold-start and hot-start—differ in their reliance on prior information when determining the  $b_{ij}$  coefficients and bias parameters.

Our previous work in [26] introduced an algorithm that optimizes both the coefficient and bias parameters to improve DC power flow accuracy across a range of operating conditions.

This optimization selects the DC power flow parameter values which minimize the mismatch between the line flows from the DC and AC power flow models over a training set of scenarios. We emphasize that the standard DC power flow parameters (e.g.,  $b_{ij} = 1/x_{ij}$ ) are themselves derived from simplifying assumptions that may not hold in practice. The goal of this parameter optimization is not to redefine the physical reactance or susceptance of a line. Rather, it is to find an effective set of parameter values that causes the DC power flow approximation to better match the results of the full AC power flow model. As discussed above, the standard DC power flow parameters are themselves derived from simplifying assumptions; we seek a different set of parameters that better serves the purpose of more accurately solving an optimization problem across a range of operating conditions.

While the previous paper [26] focused on optimizing the coefficients and biases within the DC power flow approximation itself, specialization of the parameters to specific applications can yield even larger accuracy advantages. Accordingly, this paper optimizes parameters for the DC power flow approximation in order to improve accuracy of DC-OPF solutions relative to AC-OPF solutions (as opposed to the focus on DC power flow versus AC power flow in our prior work). Specifically, we seek the DC power flow parameters that best reduce the discrepancies of the generator active power setpoints associated with the DC-OPF solutions relative to those from the AC-OPF solutions across a range of load demands. This represents a significant conceptual and technical advancement that necessitates reformulating as a bilevel optimization problem that differs substantially from the single-level problem in [26]. The upper level seeks to tune the DC-OPF parameters, while the lower level solves the convex DC-OPF problem itself. This enhanced structure introduces new challenges, as the loss function must consider the optimal operational decisions from the lower-level optimization.

Leveraging differentiable optimization techniques developed for deep learning architectures [28], [29], this paper proposes a new algorithm for adaptively selecting coefficients  $b_{ij}$  and bias parameters within the specific context of DC-OPF problems. The algorithm performs offline fine-tuning of  $b_{ij}$  and bias parameters using a training dataset of AC-OPF solutions under various representative operating conditions. We differentiate the lower-level problem to compute the sensitivities of the DC-OPF solutions with respect to these parameters. This enables us to iteratively adjust the parameter values during an offline phase by using a gradient-based optimization method, specifically the Truncated Newton Conjugate-Gradient (TNC) algorithm, to minimize the difference between DC-OPF's generator active power setpoints and their AC-OPF counterparts across the training dataset. The resulting optimized parameter values are then applied to improve the accuracy of the DC-OPF problems during online computations.

To summarize, the main contributions of this paper are:

- 1) Formulating the task of tuning DC-OPF parameters as a bilevel optimization problem, where the goal is to minimize the error in the optimal generator setpoints relative to full AC-OPF solutions.

- 2) Introducing an optimization algorithm designed to adaptively choose the coefficients  $b_{ij}$  and bias parameters of the DC power flow inside the DC-OPF problem. These parameters are instrumental in modeling line losses and incorporating factors such as shunts, HVDC infeeds, phase shifts, and line losses.
- 3) Computing gradients for the DC-OPF solution with respect to both coefficient and bias parameters to obtain sensitivities of the generator setpoints as well as our defined loss function, enabled by modern differentiable programming techniques.
- 4) Utilizing a quasi-Newton method (TNC) to scale our proposed algorithm to large power systems.
- 5) Providing numerical results that demonstrate the superior accuracy of our proposed algorithm over a range of operating conditions.

The rest of the paper is structured as follows. Section II reviews the AC-OPF and DC-OPF problems. Section III presents our algorithm for optimizing parameters of DC-OPF problems. Section IV provides numerical results to demonstrate the algorithm's performance. Section V concludes the paper.

## II. OPTIMAL POWER FLOW PROBLEM

To introduce notation and necessary background material, this section describes the AC-OPF formulation and the DC-OPF linear approximation that simplifies the AC-OPF problem to improve tractability at the cost of accuracy. Buses, lines and generators in the network are denoted by the sets  $\mathcal{N}$ ,  $\mathcal{E}$  and  $\mathcal{G}$ , respectively. Here,  $\mathcal{E}$  represents the set of individual branches; we note that multiple parallel branches may exist between two buses, and our formulation treats each one distinctly with its own parameters and flow limits. Each bus  $i \in \mathcal{N}$  has a voltage phasor  $V_i$  with phase angle  $\theta_i$ , along with a shunt admittance  $Y_i^S$ , complex power demand  $s_i^d = p_i^d + j q_i^d$ , and generated complex power  $s_i^g = p_i^g + j q_i^g$ , where  $j$  is the imaginary unit  $j = \sqrt{-1}$ . Buses without generators are modeled as having upper and lower generation limits of zero. The complex power flows entering each end of line  $(j, k) \in \mathcal{E}$  are represented by  $S_{jk}$  and  $S_{kj}$ . Series admittance parameters for each line  $(j, k)$  are  $Y_{jk}$  and  $Y_{kj}$  and the line's shunt admittance is  $Y_{jk}^c$ . The real and imaginary components of a complex number are indicated by  $\Re(\cdot)$  and  $\Im(\cdot)$ , respectively,  $\|\cdot\|_\infty$  denotes the  $L_\infty$ -norm, and  $\|\cdot\|_2$  denotes the  $L_2$ -norm. The complex conjugate and the transpose of a matrix are denoted by  $(\cdot)^*$  and  $(\cdot)^\top$ , respectively, and  $\angle(\cdot)$  is the angle of a complex argument. Upper and lower limits are represented by  $(\cdot)^+$  and  $(\cdot)^-$ .

### A. The AC-OPF Problem

Model 1 presents the AC-OPF formulation. The objective (1a) minimizes the total generation cost by summing the costs for each generator  $i \in \mathcal{N}$ , modeled as quadratic functions (with coefficients  $c_{2i}$ ,  $c_{1i}$ , and  $c_{0i}$ ) of the real part of the complex power generation  $s_i^g$ . Following typical modeling practices, we assume that  $c_{2i} \geq 0$  such that the generator cost function is convex. Constraint (1b) ensures that all voltage magnitudes are within specified limits. Constraint (1c) bounds

---

### Model 1 The AC-OPF Problem

---

$$\underset{s_i^g, V_i}{\text{minimize}} \quad \sum_{i \in \mathcal{N}} c_{2i} (\Re(s_i^g))^2 + c_{1i} \Re(s_i^g) + c_{0i} \quad (1a)$$

$$\text{s.t. } (\forall i \in \mathcal{N}, \forall (j, k) \in \mathcal{E}) \quad (1b)$$

$$\underline{V}_i \leq |V_i| \leq \bar{V}_i \quad (1b)$$

$$\underline{s}_i^g \leq s_i^g \leq \bar{s}_i^g \quad (1c)$$

$$|S_{jk}| \leq \bar{S}_{jk}, \quad |S_{kj}| \leq \bar{S}_{jk} \quad (1d)$$

$$s_i^g - s_i^d - (Y_i^s)^* |V_i|^2 = \sum_{(i,j) \in \mathcal{E}} S_{ij} + \sum_{(k,i) \in \mathcal{E}} S_{ik} \quad (1e)$$

$$S_{jk} = (Y_{jk} + Y_{jk}^c)^* V_j V_k^* - Y_{jk}^* V_j V_k \quad (1f)$$

$$S_{kj} = (Y_{kj} + Y_{kj}^c)^* V_k V_j^* - Y_{kj}^* V_k V_j \quad (1g)$$

$$\theta_{ref} = 0 \quad (1h)$$

$$-\bar{\theta}_{jk} \leq \angle(V_j V_k^*) \leq \bar{\theta}_{jk} \quad (1i)$$

$$\text{variables : } s_i^g \ (\forall i \in \mathcal{N}), V_i \ (\forall i \in \mathcal{N})$$


---

the complex power outputs of each generator within its feasible operating range, where inequalities on complex quantities are interpreted as bounds on the real and imaginary parts. Constraint (1d) enforces the line flow limits.

Constraint (1e) ensures power balance at each bus and (1f) and (1g) represent the power flows on each line. The reference angle is set to zero in (1h). Finally, (1i) limits the phase angle difference across each line. The AC-OPF problem is inherently nonlinear and non-convex due to the quadratic nature of the power flow equations. Using nonlinear optimization techniques such as interior-point methods, AC-OPF algorithms frequently find high-quality solutions [12], [13], [30].

### B. The DC-OPF Problem

The DC-OPF problem is a linear approximation of the AC-OPF problem that assumes uniform voltage magnitudes (typically normalized to one per-unit), small voltage angle differences, negligible losses, and ignores reactive power [16]. These assumptions are most relevant to transmission systems. The DC-OPF model underlies various applications within electricity markets and is widely used in unit commitment, transmission network expansion planning, and optimal transmission switching problems [9], [22], [31]–[34].

Model 2 presents the DC-OPF problem. Analogous to (1), (2a) minimizes the total generation cost, (2b) bounds the generators' active power outputs, (2c) enforces line flow limits, and (2d) balances active power at each bus. Constraint (2e) linearizes the line flow expressions by relating branch power flows to voltage angle differences according to the DC power flow approximation, thus offering a simplified counterpart to (1f) and (1g). Finally, (2f) sets the reference angle.

Note that (2d) and (2e) include parameters  $\gamma$  and  $\rho$  to improve accuracy. Following extended DC power flow models [16], these parameters have distinct roles. The flow bias  $\rho_{ij}$  in (2e) can account for factors like phase-shifter effects and active power losses on the line  $(i, j)$ . The injection bias

---

**Model 2** The DC-OPF Problem
 

---

$$\underset{p_i^g, \theta_i}{\text{minimize}} \quad \sum_{i \in \mathcal{N}} c_{2i}(p_i^g)^2 + c_{1i}p_i^g + c_{0i} \quad (2a)$$

$$\text{s.t. } (\forall i \in \mathcal{N}, \forall (j, k) \in \mathcal{E})$$

$$\underline{p}_i^g \leq p_i^g \leq \bar{p}_i^g \quad (2b)$$

$$|P_{jk}| \leq \bar{S}_{jk} \quad (2c)$$

$$p_i^g - p_i^d - \gamma_i = \sum_{(i,j) \in \mathcal{E}} P_{ij} - \sum_{(k,i) \in \mathcal{E}} P_{ki} \quad (2d)$$

$$P_{jk} = b_{jk}(\theta_j - \theta_k) + \rho_{jk} \quad (2e)$$

$$\theta_{ref} = 0 \quad (2f)$$

**variables :**  $p_i^g$  ( $\forall i \in \mathcal{N}$ ),  $\theta_i$  ( $\forall i \in \mathcal{N}$ )

---

**Model 3** The DC-OPF-M Problem
 

---

$$\underset{\mathbf{p}^g, \boldsymbol{\theta}}{\text{minimize}} \quad \mathbf{p}^g^\top \text{diag}(\mathbf{c}_2) \mathbf{p}^g + \mathbf{c}_1^\top \mathbf{p}^g + \mathbf{1}^\top \mathbf{c}_0 \quad (3a)$$

s.t.

$$\underline{\mathbf{p}}^g \leq \mathbf{p}^g \leq \bar{\mathbf{p}}^g \quad (3b)$$

$$\mathbf{p}^g - \mathbf{p}^d - \boldsymbol{\gamma} = \mathbf{A}^\top (\text{diag}(\mathbf{b}) \mathbf{A} \boldsymbol{\theta} + \boldsymbol{\rho}) \quad (3c)$$

$$|\text{diag}(\mathbf{b}) \mathbf{A} \boldsymbol{\theta} + \boldsymbol{\rho}| \leq \bar{\mathbf{S}} \quad (3d)$$

$$\theta_{ref} = 0 \quad (3e)$$

**variables :**  $\mathbf{p}^g, \boldsymbol{\theta}$

---

$\gamma_i$  in (2d) can account for losses from shunts, HVDC infeeds, or aggregate the effects of losses from all connected lines.

Model 3 equivalently rewrites Model 2 in a matrix form, where  $\mathbf{A}$  is the  $|\mathcal{E}| \times |\mathcal{N}|$  branch-bus incidence matrix and  $\text{diag}(\mathbf{b})$  is the diagonal matrix with  $\mathbf{b}$  on the diagonal.

The DC-OPF problem's accuracy is dictated by the parameters  $\mathbf{b}$ ,  $\boldsymbol{\gamma}$ , and  $\boldsymbol{\rho}$ . Traditionally, there are two DC-OPF versions, cold-start and hot-start, that rely on different amounts of prior information to select these parameters.

1) *Cold-start DC-OPF*: In this version, the coefficient and bias parameters are selected without relying on a nominal operating point. For instance, the coefficient values  $b_{ij}$  can be selected as the imaginary part of the line susceptance:

$$b_{ij}^{cold} = \Im \left( \frac{-1}{r_{ij} + jx_{ij}} \right). \quad (4)$$

We note that the alternative selection of  $b_{ij}^{cold} = 1/x_{ij}$  yields very similar numerical results to the choice of  $b_{ij}^{cold}$  in (4), so we only present results for the latter for the sake of brevity.

The cold-start version assigns values of zero to the bias parameters  $\boldsymbol{\gamma}$  and  $\boldsymbol{\rho}$ . The simplicity provided by the cold-start DC-OPF problem comes at the cost of accuracy in settings where information on a nominal operating point is available.

2) *Hot-start DC-OPF*: Hot-start DC-OPF problems leverage information from a nominal AC power flow solution to improve accuracy. There are many hot-start DC-OPF variants [16]; e.g., the “localized loss modeling” variant in [16]

selects the coefficient and bias parameters as:

$$b_{ij}^{hot} = b_{ij} |V_i^\bullet| |V_j^\bullet| \frac{\sin(\theta_i^\bullet - \theta_j^\bullet)}{\theta_i^\bullet - \theta_j^\bullet}, \quad (5a)$$

$$\gamma_i^{hot} = \sum_{(i,j) \in \mathcal{E}} \Re(Y_{ij}) |V_i^\bullet| (|V_i^\bullet| - |V_j^\bullet| \cos(\theta_i^\bullet - \theta_j^\bullet)), \quad (5b)$$

$$\rho_{ij}^{hot} = \Re(Y_{ij}) |V_i^\bullet| (|V_i^\bullet| - |V_j^\bullet| \cos(\theta_i^\bullet - \theta_j^\bullet)), \quad (5c)$$

where the notation  $(\cdot)^\bullet$  signifies quantities associated with a nominal AC power flow solution, and  $b_{ij}$  on the right-hand side of (5a) is typically the cold-start version defined in (4). The bias  $\boldsymbol{\gamma}^{hot}$  aims to capture the effects of branch losses on phase angles and  $\boldsymbol{\rho}^{hot}$  incorporates the branch losses in the line flow expressions. A notable limitation of the hot-start approach is its reliance on a single, pre-determined nominal operating point, denoted by the  $(\bullet)$  superscript in its defining equations. The accuracy of the hot-start parameters is therefore highest near that specific point and can degrade significantly as conditions deviate. This presents a challenge when a DC power flow approximation is needed for a wide range of operating conditions, as selecting hot-start parameters based on a single nominal operating point may yield insufficient accuracy.

3) *Optimized Parameters for DC Power Flow (DCPF)* [26]: Our recent work in [26] proposes an alternative for selecting the  $\mathbf{b}$ ,  $\boldsymbol{\gamma}$ , and  $\boldsymbol{\rho}$  parameters by solving a single-level optimization problem. Providing a conceptual foundation for the approach proposed in this paper, the algorithm in [26] also optimizes these parameters over a range of operation. However, there is a key distinction between the algorithm in [26] and the algorithm in this paper: the approach in [26] does not explicitly consider the impacts of the optimized parameters on the solution to a DC-OPF problem, but rather seeks to balance the DC power flow's accuracy with respect to all operating points within a specified range. Conversely, our proposed algorithm in this paper leverages machine learning techniques for training models with differentiable optimization layers to tailor the DC power flow parameter values so that the DC *optimal* power flow solutions align with the AC-OPF solutions. As shown numerically in Section IV, this enables substantial accuracy improvements relative to [26] in settings where the optimized DC power flow parameters are used in optimization problems like DC-OPF. This represents a significant conceptual and technical advancement over the single-level problem in [26]. While the prior work focused on tuning parameters to match the line flow behaviors of the DC and AC power flow *equations*, the present work introduces a bilevel optimization framework. The objective is no longer to fit the intermediate equations, but rather to tune the parameters such that the final *optimal solution* of the DC-OPF problem aligns with that of the AC-OPF problem. This solution-focused methodology is a key contribution of this paper.

We next present our algorithm for optimizing the coefficient  $(\mathbf{b})$  and bias  $(\boldsymbol{\gamma}$  and  $\boldsymbol{\rho}$ ) parameters. Our algorithm seeks to match the generators' active power outputs from the DC-OPF solution to corresponding values from the AC-OPF solution.

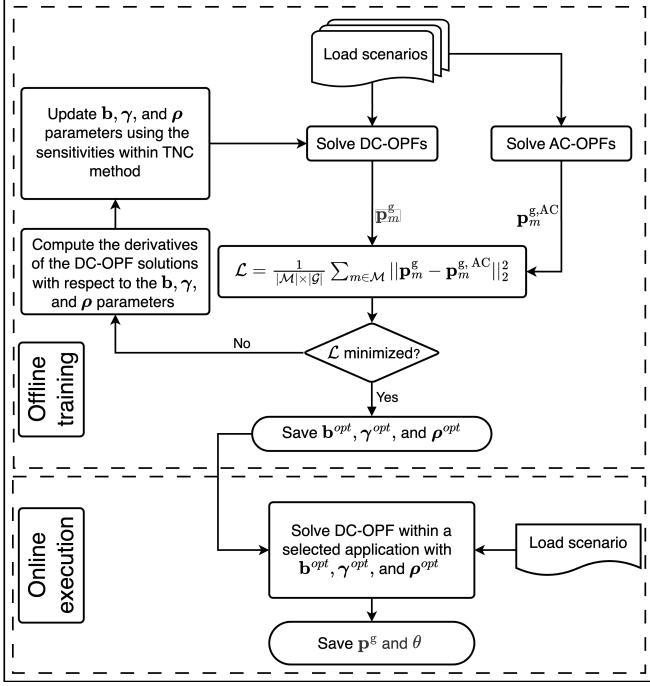


Fig. 1. Flowchart of the proposed algorithm.

### III. PARAMETER OPTIMIZATION ALGORITHM

Our parameter optimization algorithm, depicted in Fig. 1, has offline and online phases. The offline phase computes DC-OPF parameters ( $\mathbf{b}$ ,  $\gamma$ , and  $\rho$ ) to achieve a high-fidelity approximation of the AC-OPF solution over a range of operational conditions. The online phase employs these parameters to efficiently solve DC-OPF problems in various applications. While our numerical results focus on DC-OPF problems, the optimized DC-OPF parameters are applicable to many computationally intensive optimization problems that incorporate DC power flow models, such as unit commitment [9], [31], expansion planning [3], and grid reliability enhancement [35].

Our algorithm minimizes a loss function based on the discrepancy between DC-OPF and AC-OPF generator active power setpoints over various load scenarios. To perform this minimization, we compute the loss function's gradients with respect to the parameters  $\mathbf{b}$ ,  $\gamma$ , and  $\rho$  for use in a nonlinear optimization method. Based on our numerical tests comparing various optimization methods, TNC's approach to approximating the Hessian matrix from gradient information best achieved scalable and effective parameter optimization.

#### A. Loss Function

We first introduce a loss function  $\mathcal{L}$  that sums the squared two-norm discrepancies between the generators' active power setpoints from the AC-OPF problems ( $\mathbf{p}_m^{g,AC}$ ) and the DC-OPF problems ( $\mathbf{p}_m^g$ ) across a specified set of load scenarios  $\mathcal{M}$ :

$$\mathcal{L}(\mathbf{b}, \gamma, \rho) = \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \|\mathbf{p}_m^g(\mathbf{b}, \gamma, \rho) - \mathbf{p}_m^{g,AC}\|_2^2, \quad (6)$$

where the constant  $\frac{1}{|\mathcal{G}| \times |\mathcal{M}|}$  normalizes this function based on the number of generators and load scenarios. As shown in (6),  $\mathbf{p}_m^g$  (and thus  $\mathcal{L}(\mathbf{b}, \gamma, \rho)$ ) is a function of the coefficient

#### Model 4 Optimizing DC-OPF Problem

$$\underset{\mathbf{b}, \gamma, \rho}{\text{minimize}} \quad \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \|\mathbf{p}_m^g(\mathbf{b}, \gamma, \rho) - \mathbf{p}_m^{g,AC}\|_2^2 \quad (8a)$$

s.t.

$$\underset{\mathbf{p}_m^g, \theta_m}{\text{minimize}} \quad \mathbf{p}_m^g^\top \text{diag}(\mathbf{c}_2) \mathbf{p}_m^g + \mathbf{c}_1^\top \mathbf{p}_m^g + \mathbf{1}^\top \mathbf{c}_0 \quad (8b)$$

s.t.

$$\mathbf{p}_m^g - \mathbf{p}_m^d - \gamma = \mathbf{A}^\top (\text{diag}(\mathbf{b}) \mathbf{A} \theta_m + \rho) \quad (8c)$$

$$|\text{diag}(\mathbf{b}) \mathbf{A} \theta_m + \rho| \leq \bar{\mathbf{S}} \quad (8d)$$

$$\mathbf{p}_m^g \leq \mathbf{p}_m^g \leq \bar{\mathbf{p}}^g \quad (8e)$$

$$\theta_{ref} = 0 \quad (8f)$$

$$\forall m \in \mathcal{M}$$

parameters  $\mathbf{b}$  and the bias parameters  $\gamma$  and  $\rho$ . The selection of the training scenarios in the set  $\mathcal{M}$  is an important application-dependent step. Scenarios should be chosen to represent the operating conditions under which the optimized parameters will be used. In our experiments, we generate scenarios by scaling nominal loads based on a normal distribution; however, in practice, utilities could use historical data or forecast scenarios to create a training set tailored to their system.

We note that this two-norm loss formulation is typical in machine learning for its robustness and differentiability. Moreover, this loss function penalizes larger deviations more heavily, which is well aligned with typical applications where a small number of severe approximation errors would be more problematic than a large number of minor errors. One could instead use other norms without major conceptual changes.

With this loss function, the optimization problem to find the best coefficient and bias parameters is formulated as:

$$\underset{\mathbf{b}, \gamma, \rho}{\text{minimize}} \quad \mathcal{L}(\mathbf{b}, \gamma, \rho). \quad (7)$$

Using the sensitivity analysis that we describe in the next subsection, we adjust the coefficient and bias parameters ( $\mathbf{b}$ ,  $\gamma$ , and  $\rho$ ) to optimize the DC-OPF problem's parameters. As detailed in Model 4, this task takes the form of a bilevel optimization problem, with the upper-level fine-tuning parameters based on outcomes from lower-level DC-OPF problems under a variety of operational scenarios.

#### B. Sensitivity Analysis of Coefficient and Bias Parameters

This subsection describes the sensitivity analyses crucial for our parameter optimization algorithm. We utilize a gradient-based method, TNC, which requires the gradient of the loss function with respect to  $\mathbf{b}$ ,  $\gamma$ , and  $\rho$ . These gradients, denoted as  $\mathbf{g}^b$ ,  $\mathbf{g}^\gamma$ , and  $\mathbf{g}^\rho$ , are computed as follows:

$$\mathbf{g}^b = \frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \left( \frac{\partial \mathbf{p}_m^g}{\partial \mathbf{b}} \Big|_{\mathbf{p}_m^g} \right)^\top \left( \mathbf{p}_m^g - \mathbf{p}_m^{g,AC} \right), \quad (9a)$$

**Algorithm 1:** Truncated Newton (TNC) Method

---

**Input:**  $\mathbf{x}_0 = [\mathbf{b}_0^\top, \boldsymbol{\gamma}_0^\top, \boldsymbol{\rho}_0^\top]^\top$ : Initial guess  
 $\epsilon$ : Tolerance for convergence  
 $max\_iter$ : Maximum iterations  
 $\mathcal{L}(\mathbf{x}_k)$ : Loss function  
 $\nabla \mathcal{L}(\mathbf{x}_k) = \mathbf{g}$   
 $\mathbf{M}$ : Preconditioning matrix (often diagonal)  
 $\mathbf{H}$ : Hessian or its approximation function  
 $\alpha_1$ : Armijo condition constant (e.g.,  $10^{-4}$ )  
 $\alpha_2$ : Curvature condition constant, between  $\alpha_1$  and 1

**Output:** Optimized parameters  $\mathbf{x}^*$

```

1 Initialize  $\mathbf{x}_k \leftarrow \mathbf{x}_0$ ,  $k \leftarrow 0$ 
2 while  $k \leq max\_iter$  and  $\|\nabla \mathcal{L}(\mathbf{x}_k)\| > \epsilon$  do
3    $\mathbf{g} \leftarrow \nabla \mathcal{L}(\mathbf{x}_k)$ 
4    $\mathbf{r} \leftarrow -\mathbf{g}$ 
5    $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$ 
6    $\mathbf{p} \leftarrow \mathbf{z}$ 
7    $\mathbf{s} \leftarrow \mathbf{0}$ 
8    $\rho_{old} \leftarrow \mathbf{r}^\top \mathbf{z}$ 
9   while  $\|\mathbf{r}\| > \epsilon$  do
10     $\mathbf{q} \leftarrow \mathbf{H}\mathbf{p}$ 
11     $\alpha \leftarrow \frac{\rho_{old}}{\mathbf{p}^\top \mathbf{q}}$ 
12     $\mathbf{s} \leftarrow \mathbf{s} + \alpha \mathbf{p}$ 
13     $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathbf{q}$ 
14     $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$ 
15     $\rho_{new} \leftarrow \mathbf{r}^\top \mathbf{z}$ 
16     $\eta \leftarrow \frac{\rho_{new}}{\rho_{old}}$ 
17     $\mathbf{p} \leftarrow \mathbf{z} + \eta \mathbf{p}$ 
18     $\rho_{old} \leftarrow \rho_{new}$ 
19    // Wolfe Line Search to determine  $\beta$ 
20     $\beta \leftarrow 1$ 
21    while True do
22      if  $\mathcal{L}(\mathbf{x}_k + \beta \mathbf{s}) \leq \mathcal{L}(\mathbf{x}_k) + \alpha_1 \beta \mathbf{g}^\top \mathbf{s}$  and
          $\|\nabla \mathcal{L}(\mathbf{x}_k + \beta \mathbf{s})^\top \mathbf{s}\| \leq \alpha_2 \|\mathbf{g}^\top \mathbf{s}\|$  then
        break
23       $\beta \leftarrow \beta/2$ 
24     $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \beta \mathbf{s}$ 
25     $k \leftarrow k + 1$ 
26  $\mathbf{x}^* \leftarrow \mathbf{x}_k$ 

```

---

$$\mathbf{g}^\gamma = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\gamma}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \left( \frac{\partial \mathbf{p}_m^g}{\partial \boldsymbol{\gamma}} \Big|_{\mathbf{p}_m^g} \right)^\top (\mathbf{p}_m^g - \mathbf{p}_m^{g, AC}), \quad (9b)$$

$$\mathbf{g}^\rho = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \left( \frac{\partial \mathbf{p}_m^g}{\partial \boldsymbol{\rho}} \Big|_{\mathbf{p}_m^g} \right)^\top (\mathbf{p}_m^g - \mathbf{p}_m^{g, AC}). \quad (9c)$$

The Jacobian matrices  $\frac{\partial \mathbf{p}_m^g}{\partial \mathbf{b}}$ ,  $\frac{\partial \mathbf{p}_m^g}{\partial \boldsymbol{\gamma}}$ , and  $\frac{\partial \mathbf{p}_m^g}{\partial \boldsymbol{\rho}}$  are central to the gradients in (9). Since  $\mathbf{p}_m^g$  is the solution to the convex DC-OPF optimization problem, these derivatives are found by implicitly differentiating the Karush-Kuhn-Tucker (KKT) optimality conditions of the DC-OPF problem [36], [37]. This is a key step that makes our bilevel approach tractable.

To perform these implicit derivative computations, we leverage techniques developed for embedding convex optimization layers within deep learning architectures; specifically, we utilize `cvxpylayers` [38], which allows for automatic differentiation through convex optimization problems specified in CVXPY. This integration facilitates efficient and scalable

gradient computations necessary for our optimization framework. Our implementation defines the DC-OPF problem using CVXPY's modeling syntax and uses `cvxpylayers` to create a differentiable layer. During the forward pass, this layer solves the DC-OPF problem to obtain  $\mathbf{p}_m^g$ . In the backward pass, it automatically computes the required Jacobians by differentiating the KKT system, leveraging the methods in [37], [38].

Next, we will use the collected gradient information defined as  $\nabla \mathcal{L}(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho}) = \mathbf{g} = [\mathbf{g}^b^\top, \mathbf{g}^\gamma^\top, \mathbf{g}^\rho^\top]^\top$  in the TNC optimization method to update the coefficient and bias parameters.

### C. Optimization Formulation and Solution Method

To solve the bilevel problem in Model 4, we employ the gradient-based TNC optimization method that uses the calculated parameter sensitivities [39], [40]. It is important to clarify that TNC is used to solve the non-convex upper-level problem (7) of finding the optimal parameters  $(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho})$ . The lower-level DC-OPF problem for each scenario is solved using the Gurobi optimizer, which is invoked through the `cvxpylayers` library. TNC combines the advantages of Newton's method for fast convergence with the conjugate gradient approach to efficiently handle large-scale problems without the need for explicit Hessian computations. When numerically compared to other common gradient-based optimizers, including Broyden-Fletcher-Goldfarb-Shanno (BFGS) and limited-memory BFGS [40], we found that TNC consistently demonstrated superior convergence speed and reliability for our specific problem structure, also shown in [26].

As shown in Algorithm 1, TNC iteratively updates the solution vector by approximating the Newton direction using truncated conjugate gradients with preconditioning. This begins with an initial guess for the solution vector, followed by preparing an initial diagonal approximation to the Hessian matrix. Each iteration computes the gradient of the loss function and applies the conjugate gradient method to approximate the Newton direction while ensuring that the preconditioner improves numerical stability and accelerates convergence. The stopping criterion for the inner conjugate gradient loop is based on the reduction of the gradient's magnitude. The optimal step length along the search direction is then determined using a line search that satisfies the Wolfe conditions, ensuring sufficient decrease in the objective function. The solution vector is updated accordingly, and the process repeats until the outer convergence criteria are satisfied.

Note that we use a standard implementation of the TNC algorithm to solve the bilevel optimization problem in Model 4. Other optimization methods could be used without conceptual changes. The novelty of this work lies in the formulation and overall solution approach for the bilevel problem, rather than the application of TNC itself. For a more detailed discussion on the standard TNC method, refer to [39].

As an alternative solution approach, we could have reformulated the bilevel optimization problem by replacing the lower-level problem with its KKT conditions to obtain a single-level problem [41]. Despite its theoretical elegance, this single-level reformulation presents several practical difficulties that can render it intractable. The simultaneous consideration of multiple scenarios yields a large-scale nonlinear program that is

computationally demanding. Furthermore, this reformulation introduces non-convexities associated with the bilinear terms from the complementary slackness and stationarity conditions, yielding a mathematical program with equilibrium constraints (MPEC) [42]. Due to challenges associated with the resulting constraint qualification conditions, this reformulation complicates the solution process, necessitating the use of specialized nonlinear programming solvers whose convergence is notoriously sensitive to various hyperparameters. Thus, for the sake of tractability, we do not explicitly enforce KKT-based constraints. Instead, we follow typical approaches for training machine learning models that rely on sensitivities to obtain gradient-based algorithms that scale well in related settings. Gradients of the loss function with respect to the parameters ( $\mathbf{b}, \gamma, \rho$ ) are obtained by implicitly differentiating the KKT conditions of the lower-level DC-OPF problem, leveraging modern differentiable optimization libraries like cvxpylayers.

#### IV. NUMERICAL EXPERIMENTS

Using systems from the PGLib-OPF test case archive [43], including IEEE 14-bus, EPRI 39-bus, IEEE 57-bus, IEEE 118-bus, Illinois 200-bus, and Texas 2000-bus [43], this section numerically evaluates our algorithm's performance via comparison against traditional DC-OPF and AC-OPF models.

For each system, we generated 2,020 load scenarios, allocating 20 for offline training and 2,000 for testing, by multiplicatively scaling the nominal loads with normally distributed random variables, each load independently by its own variable, with mean  $\mu = 1$  and standard deviation  $\sigma = 15\%$ . The active and reactive demands at each bus were scaled by the same multiplicative factor to keep the loads' nominal power factors. To ensure a valid basis for comparison, only load scenarios that resulted in a feasible AC-OPF solution were included in the training and testing datasets.

All computations were conducted on a 24-core, 32 GB RAM computing node at Georgia Tech's Partnership for an Advanced Computing Environment (PACE). The AC-OPF problems were solved using PowerModels.jl [44]. We implemented our algorithm in PyTorch with derivatives  $\frac{\partial \mathbf{P}^g}{\partial \mathbf{b}}$ ,  $\frac{\partial \mathbf{P}^g}{\partial \gamma}$ , and  $\frac{\partial \mathbf{P}^g}{\partial \rho}$  calculated via cvxpylayers [38]. The TNC method from the `scipy.optimize.minimize` library was used to minimize the loss function (6) using the sensitivities  $\mathbf{g}^b$ ,  $\mathbf{g}^\gamma$ , and  $\mathbf{g}^\rho$ .

We benchmark the accuracy achieved by our optimized parameters by comparing the generators' active power set-points across different DC-OPF models (i.e.,  $\mathbf{p}^{[\text{model}]}$  where  $[\text{model}]$  stands for either the optimized parameters from our proposed algorithm, the cold-start DC-OPF (4), the hot-start DC-OPF (5), or the optimized DC power flow (DCPF) parameters from [26]) against the true values from the AC-OPF solutions (i.e.,  $\mathbf{p}^{g,\text{AC}}$ ). For the hot-start DC-OPF benchmark, the parameters are derived from an AC power flow solution at the nominal loads (i.e., the base case with a 1.0 scaling factor), which represents the center of the distribution used for scenario generation. The discrepancies are measured using both the mean square error (MSE) as defined in the loss

function (6) and the maximum error metric, with all values in the per unit (p.u.) system with a 100 MVA base power:

$$\varepsilon_{\text{MSE}}^{[\text{model}]} = \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \|\mathbf{p}_m^{[\text{model}]} - \mathbf{p}_m^{g,\text{AC}}\|_2^2, \quad (10)$$

$$\varepsilon_{\max}^{[\text{model}]} = \max_{m \in \mathcal{M}} \|\mathbf{p}_m^{[\text{model}]} - \mathbf{p}_m^{g,\text{AC}}\|_\infty, \quad (11)$$

where  $\|\cdot\|_\infty$  is the  $L_\infty$ -norm and  $\|\cdot\|_2$  is the  $L_2$ -norm.

#### A. Comparison of Parameter Values Across Selection Methods

This section presents the distributions of the cold-start, hot-start, and optimized parameters  $\mathbf{b}$ ,  $\gamma$ , and  $\rho$  for different test systems using box plots and scatter plots. The box plots show the spread of parameter values across the buses and lines in the system, providing insight into how the parameter values vary within the system. The scatter plots compare the hot-start and optimized parameters for each bus or line, demonstrating the consistency of the optimized parameters with those used in a common DC power flow approximation.

1) *Box Plot Analysis*: Fig. 2 illustrates the distribution of the parameter values using box plots. The interquartile range (IQR), which represents the middle 50% of the data, is shown in each box, with the central line marking the median value. Whiskers extend to 1.5 times the IQR, and data points outside this range are considered outliers, marked individually. The box plots for each test system capture the cold-start values ( $\mathbf{b}^{cold}$ ), hot-start values ( $\mathbf{b}^{hot}$ ,  $\gamma^{hot}$ , and  $\rho^{hot}$ ), and the optimized values ( $\mathbf{b}^{opt}$ ,  $\gamma^{opt}$ , and  $\rho^{opt}$ ). All values are presented on a logarithmic scale to accommodate variations across multiple orders of magnitude. The box plots in Fig. 2 show that the optimized parameter values closely align with those derived from traditional heuristics, particularly for the coefficient parameters. This suggests consistency with long-standing power engineering intuition, where line susceptances are known to be a key factor in determining power flows. While the optimization allows negative  $b$  values, all of the optimized  $b$  values were positive for all of the test cases. This is consistent with the underlying line reactances in these systems, all of which are positive. If there were lines with negative reactances (e.g., series-compensated lines), the method would yield negative  $b$  values if this would improve accuracy.

2) *Scatter Plot Analysis*: Complementing the box plots, the scatter plots (Fig. 3) directly compare the hot-start and optimized parameter values. Each plot includes a  $45^\circ$  red dashed line to indicate one-to-one correspondence. Data points falling along this line suggest a close match between the hot-start and optimized values, while deviations highlight discrepancies. Fig. 3a compares the coefficient values  $\mathbf{b}^{hot}$  and  $\mathbf{b}^{opt}$ , while Fig. 3b compares the injection bias values  $\gamma^{hot}$  and  $\gamma^{opt}$ , and Fig. 3c compares the flow bias values  $\rho^{hot}$  and  $\rho^{opt}$ . These scatter plots reveal that the optimized parameters align well with hot-start values, except for a few outliers where targeted adjustments to  $\mathbf{b}$ ,  $\gamma$ , and  $\rho$  improve the DC-OPF approximation's accuracy. While Fig. 3 shows general alignment between optimized and hot-start parameters, we emphasize that this does not imply that the optimized parameters yield marginal accuracy improvements. Hot-start

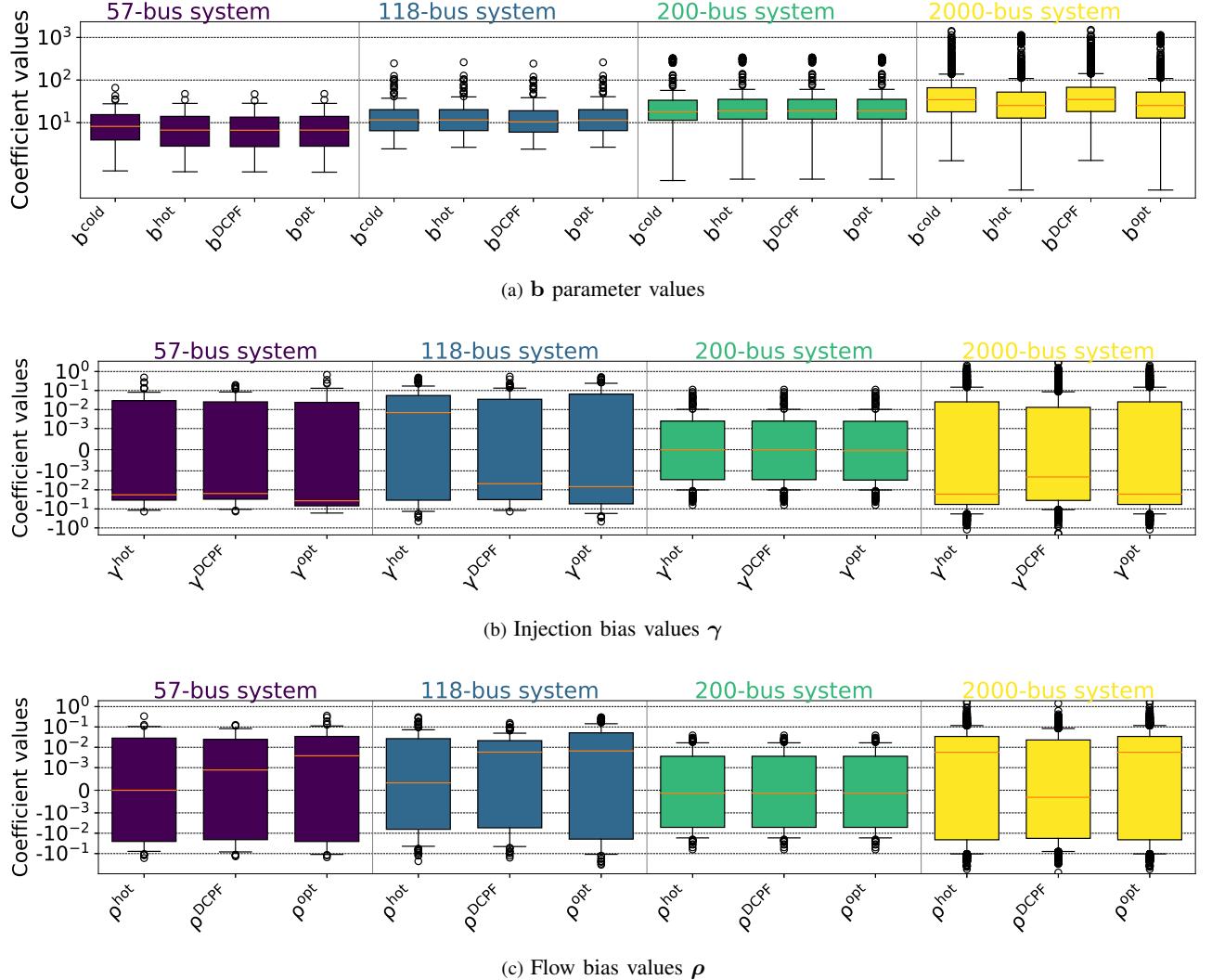


Fig. 2. Boxplots showing the distributions of the parameter values ( $\mathbf{b}$ ,  $\gamma$ , and  $\rho$ ) across various test cases for cold-start, hot-start, DCPF, and optimized parameters.

is a strong heuristic, and our algorithm's ability to learn small but systematic deviations from it is precisely what enables substantial accuracy gains across diverse scenarios, as we discuss next.

#### B. Comparison of Approximation Accuracies

Table I gives the squared two-norm and  $\infty$ -norm errors for several test cases when using the optimized and conventional DC-OPF parameters over 2,000 testing scenarios. Notably, in the 118-bus test case, our algorithm achieved a substantial reduction in the squared two-norm loss to 0.0123, compared to the cold-start, hot-start, and optimized DCPF parameters with 0.1295, 0.1221, and 0.1251, respectively. This represents an order of magnitude improvement over all prior methods, highlighting the effectiveness of the proposed algorithm. While the absolute  $\infty$ -norm error remains large in some cases, these outliers typically correspond to stressed conditions that any DC power flow approximation struggles to model. Our method cannot fully eliminate all approximation errors, but nevertheless significantly reduces the worst-case error (by as

much as 79% across test cases) compared to traditional DC-OPF parameterizations. This shows that the proposed approach substantially increases the DC power flow approximation's accuracy while preserving its computational advantages.

The improvement column in Table I shows the percentage improvement from the optimized parameters in the DC-OPF models compared to the cold-start, hot-start, and optimized DCPF parameters. The results show up to 90% and 79% improvements in the squared two-norm and  $\infty$ -norm errors.

The box plot in Fig. 4 compares the mean generator outputs for the IEEE 118-bus system across five formulations: cold-start, hot-start, DCPF, DC-OPF with optimal parameters, and AC-OPF. The vertical axis represents the generator output in per unit, and the horizontal axis shows the different parameter choices. Each box illustrates the distribution of mean generator outputs for non-zero generators across 2,000 test scenarios, highlighting the variability and median for each method. The figure shows that the DC-OPF with optimized parameters ( $\mathbf{b}^{opt}$ ,  $\gamma^{opt}$ , and  $\rho^{opt}$ ) aligns closely with the AC-OPF, as indicated by similar medians and ranges.

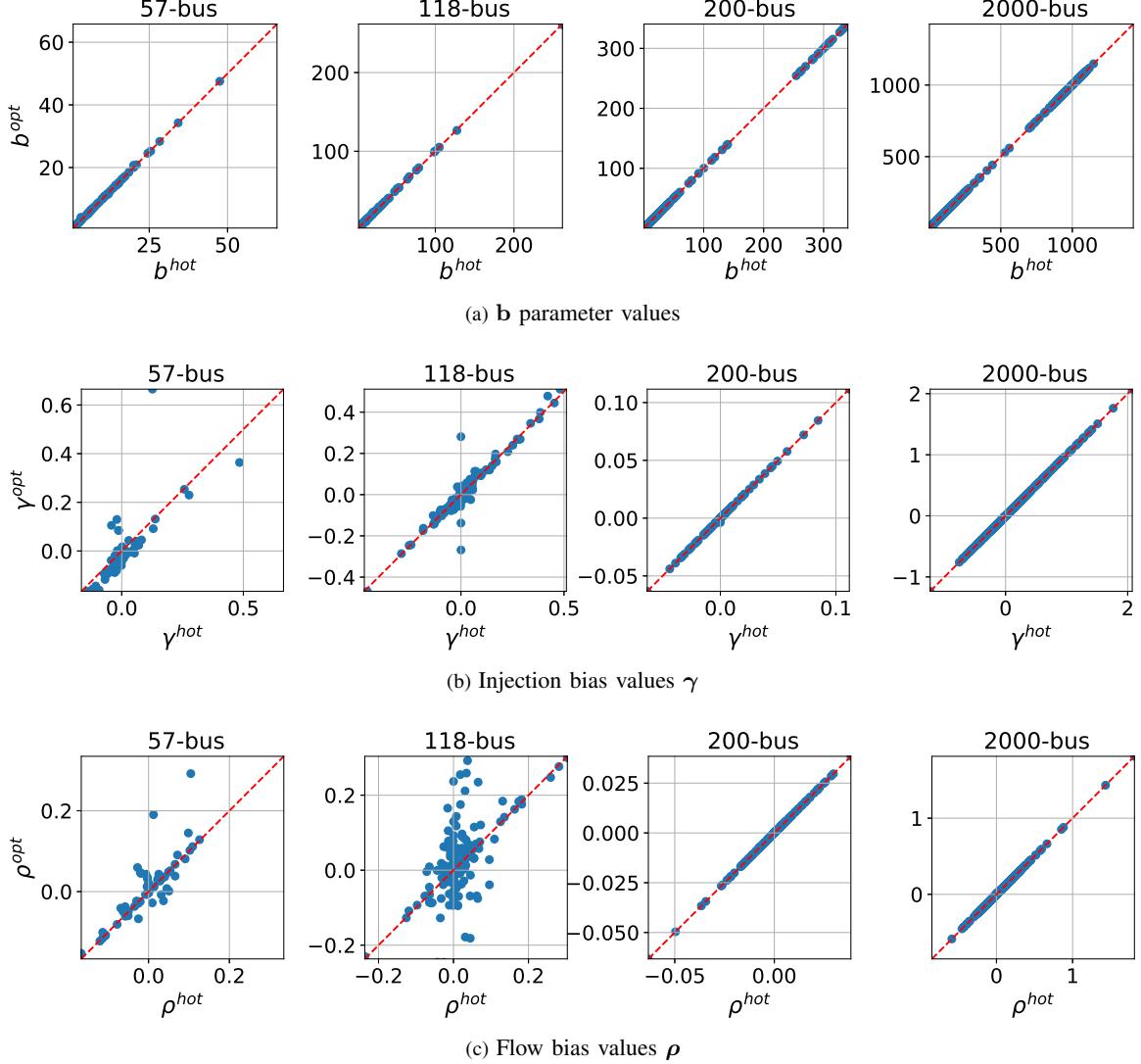


Fig. 3. Scatter plots comparing hot-start ( $\mathbf{b}^{hot}$ ,  $\gamma^{hot}$ ,  $\rho^{hot}$ ) and optimized ( $\mathbf{b}^{opt}$ ,  $\gamma^{opt}$ ,  $\rho^{opt}$ ) parameter values across various test cases. The red dashed line indicates a one-to-one correspondence.

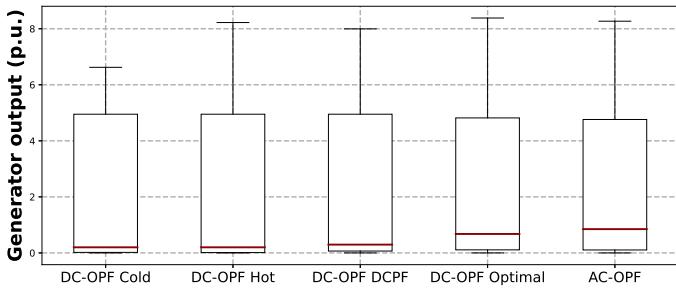


Fig. 4. Comparison of the generator outputs for the IEEE 118-bus system over 2,000 test scenarios. The box plot compares the performance of the AC-OPF vs DC-OPF with four parameter sets: cold-start ( $\mathbf{b}^{cold}$ ,  $\gamma^{cold}$ , and  $\rho^{cold}$ ), hot-start ( $\mathbf{b}^{hot}$ ,  $\gamma^{hot}$ , and  $\rho^{hot}$ ), DCPF-based parameters ( $\mathbf{b}^{DCPF}$ ,  $\gamma^{DCPF}$ , and  $\rho^{DCPF}$ ), and optimized parameters ( $\mathbf{b}^{opt}$ ,  $\gamma^{opt}$ , and  $\rho^{opt}$ ).

The graphical analyses provided in Figs. 5 and 6 offer further insights into the performance of the proposed algorithm. In Fig. 5, the generator setpoints for the IEEE 118-

bus test system are presented, where we compare the AC-OPF solutions (i.e.,  $\mathbf{p}^{g,AC}$ ) with the DC-OPF solutions derived from models with cold-start, hot-start, DCPF, and optimized DC-OPF parameters. Fig. 5a displays the setpoints of individual generators, excluding those that consistently reached their maximum output limits across all parameter choices. The alignment of setpoints with the 45-degree equality line demonstrates how closely the DC-OPF models, particularly the one with optimized parameters, mimic the AC-OPF results. Fig. 5b shows the average generator setpoints (p.u.) with standard deviation ellipses across 2,000 scenarios. These ellipses illustrate the variability and performance of each DC-OPF method, highlighting the consistency of the optimized DC-OPF in closely matching the AC-OPF setpoints. The inset focuses on generators with lower setpoints, providing a detailed view of their alignment and variability. These results demonstrate that our optimized DC-OPF parameters achieve setpoints comparable to the more complex AC-OPF while offering enhanced computational efficiency.

TABLE I  
SQUARED TWO-NORM AND  $\infty$ -NORM LOSS FUNCTIONS USING 2,000 TEST SCENARIOS

Test case	Squared Two-Norm Loss					$\infty$ -Norm Loss				
	$\varepsilon_{\text{MSE}}^{\text{cold}}$	$\varepsilon_{\text{MSE}}^{\text{hot}}$	$\varepsilon_{\text{MSE}}^{\text{DCPF}} [26]$	$\varepsilon_{\text{MSE}}^{\text{opt}}$	Improv. (%)	$\varepsilon_{\text{max}}^{\text{cold}}$	$\varepsilon_{\text{max}}^{\text{hot}}$	$\varepsilon_{\text{max}}^{\text{DCPF}} [26]$	$\varepsilon_{\text{max}}^{\text{opt}}$	Improv. (%)
14-bus	0.0070	0.0069	0.0069	<b>0.0030</b>	(57, 57, 57)	0.590	0.590	0.590	0.590	(0, 0, 0)
39-bus	0.3223	0.4248	0.3046	<b>0.3029</b>	(6, 28, 1)	5.877	6.215	5.784	<b>5.585</b>	(5, 10, 3)
57-bus	0.7445	0.6311	0.2054	<b>0.1765</b>	(76, 72, 14)	4.671	4.590	3.658	<b>3.120</b>	(33, 32, 15)
118-bus	0.1295	0.1221	0.1251	<b>0.0123</b>	(90, 90, 90)	3.133	3.170	3.146	<b>1.918</b>	(39, 39, 39)
200-bus	0.0031	0.0003	0.0003	<b>0.0002</b>	(93, 33, 33)	0.145	0.033	0.035	<b>0.031</b>	(79, 6, 11)
500-bus	0.0073	0.0069	0.0054	<b>0.0030</b>	(59, 56, 44)	1.356	1.000	0.950	<b>0.890</b>	(34, 11, 6)
2000-bus	0.0149	0.0161	0.0156	<b>0.0069</b>	(54, 57, 56)	0.829	1.024	0.967	<b>0.623</b>	(25, 39, 36)

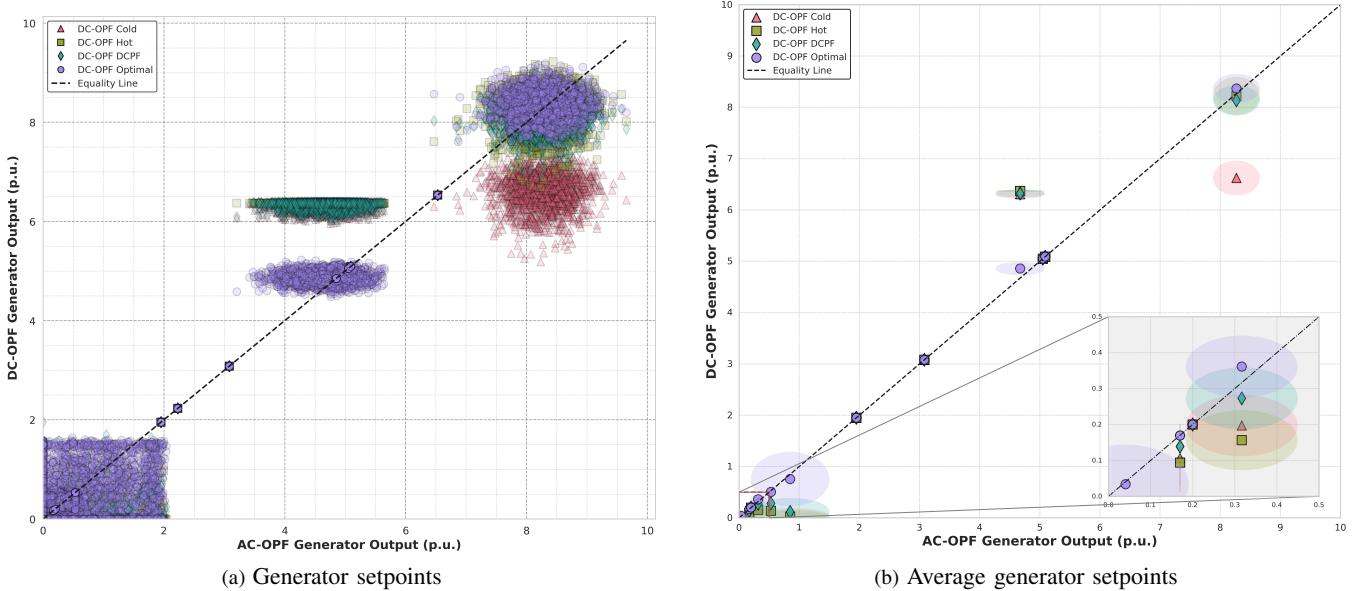


Fig. 5. (a) Generator setpoints (p.u.) obtained by solving the AC-OPF problem and DC-OPF problems with cold-start, hot-start, DCPF, and optimized parameters for the IEEE 118-bus test system. Generators that consistently reached their maximum limits across all models are excluded. The 45-degree equality line indicates perfect alignment between the AC-OPF and DC-OPF setpoints, serving as a benchmark for comparison. (b) Average generator setpoints (p.u.) with standard deviation ellipses across 2,000 scenarios.

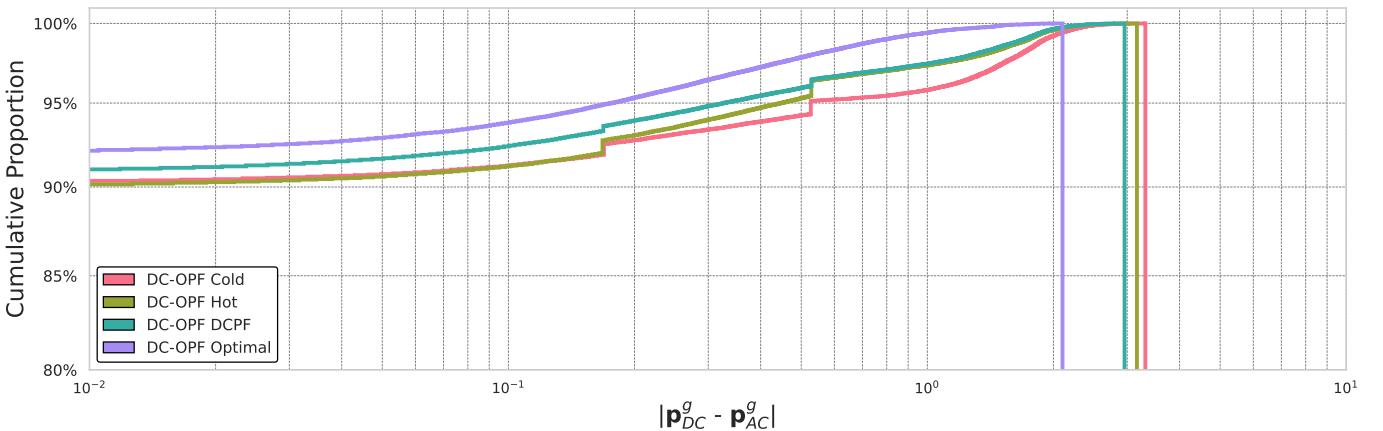


Fig. 6. Cumulative proportion of the absolute error between AC-OPF and DC-OPF (in per unit) for the IEEE 118-bus system over 2,000 test scenarios. The graph compares four scenarios: usage of cold-start ( $\mathbf{b}^{\text{cold}}$ ,  $\gamma^{\text{cold}}$ , and  $\rho^{\text{cold}}$ ), hot-start ( $\mathbf{b}^{\text{hot}}$ ,  $\gamma^{\text{hot}}$ , and  $\rho^{\text{hot}}$ ), DCPF ( $\mathbf{b}^{\text{DCPF}}$ ,  $\gamma^{\text{DCPF}}$ , and  $\rho^{\text{DCPF}}$ ), and optimized ( $\mathbf{b}^{\text{opt}}$ ,  $\gamma^{\text{opt}}$ , and  $\rho^{\text{opt}}$ ) parameters, shown on a logarithmic scale.

TABLE II  
CPU TIME IN SECONDS

Test case	14-bus	39-bus	57-bus	118-bus	200-bus	500-bus	2000-bus
$t_{train}$	13	17	41	52	58	903	9357
$t_{DCOPF}$	0.009	0.021	0.024	0.064	0.055	0.354	1.304
$t_{ACOPF}$	0.030	0.071	0.091	0.295	0.289	1.650	8.708

$t_{train}$ : Offline computation time to train the parameters.

$t_{DCOPF}$ : Online per-scenario computation time averaged over 100 scenarios.

$t_{ACOPF}$ : Online per-scenario computation time averaged over 100 scenarios.

Fig. 6 shows the cumulative proportion of the absolute errors between AC-OPF and DC-OPF solutions for the IEEE 118-bus system. The graph compares four scenarios: usage of cold-start  $\mathbf{b}$ , hot-start  $\mathbf{b}^{hot}$ ,  $\gamma^{hot}$ ,  $\rho^{hot}$ , DCPF  $\mathbf{b}^{DCPF}$ ,  $\gamma^{DCPF}$ ,  $\rho^{DCPF}$ , and optimized  $\mathbf{b}^{opt}$ ,  $\gamma^{opt}$ ,  $\rho^{opt}$  parameters, shown on a logarithmic scale. Points located towards the upper left represent better performance, indicating a larger proportion of smaller errors, which demonstrates the improved accuracy of the DC-OPF solutions using the optimized parameters.

Observe that the maximum errors are less than 1.918 per unit for the solution obtained from the optimized DC-OPF model versus errors up to 3.133, 3.170, and 3.146 per unit resulting from the cold-start DC-OPF with  $\mathbf{b}^{cold}$ , the hot-start DC-OPF with  $\mathbf{b}^{hot}$ ,  $\gamma^{hot}$ , and  $\rho^{hot}$ , and the DCPF with  $\mathbf{b}^{DCPF}$ ,  $\gamma^{DCPF}$ , and  $\rho^{DCPF}$ , respectively.

Finally, Table II summarizes the training, testing, and AC-OPF model computation times. Training times, denoted as  $t_{train}$ , range from 13 to 9,357 seconds. This represents a one-time, upfront effort that is within reasonable ranges for offline computations (minutes to hours for larger systems). We note that the primary bottlenecks in training computations are solving the DC-OPF problems and computing the gradients of the loss function with respect to the parameters. On average, solving the DC-OPF problems accounts for approximately 45% of the total training time, while computing the gradients contributes to 55% of the training time. We note that an implementation that utilizes a stochastic gradient descent method, as is often employed when training machine learning models, could reduce both components of the training time.

Post-training, the solution times for the DC-OPF model ( $t_{DCOPF}$ ) range from 0.009 to 1.304 seconds per scenario, significantly faster than the AC-OPF model due to the simplicity of solving linear programs versus nonlinear programs. Moreover, the variation in DC power flow parameters does not impact DC-OPF solution times, which remain consistent to within 2% across different parameter choices.

While the times in Table II show that a single AC-OPF solve is computationally acceptable in some contexts, the key advantages of a fast and accurate DC-OPF model emerge in more complex applications. Problems such as unit commitment, optimal transmission switching, or stochastic optimization often involve solving many thousands of OPF subproblems [2], [8], [11]. For these applications, AC-OPF computational costs are often prohibitive, making a high-accuracy DC-OPF model indispensable. The proposed algorithm provides parameters that enhance DC-OPF accuracy for these computationally intensive tasks without sacrificing its essential speed advantage.

Compared to our prior work optimizing parameters to match DC and AC *power flows* [26], the bilevel approach in this paper is computationally more intensive since differentiating through a constrained optimization problem (DC-OPF) is more complex than differentiating through the closed-form power flow equations. The benefits of the method's enhanced accuracy outweigh its additional computational cost. As shown in Table I, directly optimizing for the end-goal—accurate generator setpoints from an OPF problem—yields significantly greater accuracy improvements (e.g., a 90% vs. 14% improvement in MSE for the 57-bus case when comparing the proposed method to the DCPF method of [26]). This superior performance justifies the added one-time offline computational effort for applications where solution quality is paramount.

## V. CONCLUSION

This paper presented an optimization algorithm for significantly enhancing the accuracy of DC-OPF problems by selecting coefficient and bias parameters in the DC power flow approximation based on a sensitivity analysis. Employing the TNC method for optimization alongside efficient gradient computation techniques, an offline training phase refines DC power flow parameters across a range of loading scenarios. With squared two-norm and  $\infty$ -norm losses of up to 90% and 79%, respectively, lower than traditional parameter selection methods for DC-OPF problems, our numerical tests demonstrate a substantial improvement in DC-OPF solution accuracy relative to the AC-OPF problem.

Future work will evaluate the optimized parameters in complex applications like unit commitment and infrastructure hardening. We will also test their generalization to security-constrained problems, including  $N - 1$  contingencies, and their robustness against more diverse daily and seasonal load profiles. Our preliminary experience suggests that achieving the most substantial improvements will likely require adapting the bilevel formulation and loss function to the specific objectives and constraints of each application (e.g., discrete decisions and multi-period constraints in unit commitment). As the complexity of the training data increases—for example, by including generator commitment variations—the optimization landscape may become more challenging. While TNC proved effective in this study, future implementations can easily adopt other optimizers through the same `scipy.optimize` interface to assess their performance in other applications. Options such as stochastic gradient-based or hybrid methods are standard for large-scale, non-convex problems and could further improve the speed and reliability of convergence under more realistic and complex operational scenarios. Finally, we plan to analyze the underlying system characteristics, such as network congestion and topology, to determine the conditions under which our method provides the most significant accuracy gains.

## REFERENCES

- [1] M. B. Cain, R. P. O'Neill, and A. Castillo, "History of optimal power flow and formulations (OPF Paper 1)," *Federal Energy Regulatory Commission*, December 2012.

- [2] 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," *Elect. Power Syst. Res.*, vol. 214, p. 108725, 2023, presented at the 22nd Power Syst. Comput. Conf. (PSCC).
- [3] M. Majidi-Qadikolai and R. Baldick, "Stochastic transmission capacity expansion planning with special scenario selection for integrating  $N - 1$  contingency analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4901–4912, 2016.
- [4] S. Abhyankar, G. Geng, M. Anitescu, X. Wang, and V. Dinavahi, "Solution Techniques for Transient Stability-Constrained Optimal Power Flow – Parts I and II," *IET Generation, Transmission & Distribution*, vol. 11, pp. 3177–3193, August 2017.
- [5] D. Bienstock and A. Verma, "Strong NP-hardness of AC power flows feasibility," *Oper. Res. Lett.*, vol. 47, no. 6, pp. 494–501, 2019.
- [6] D. K. Molzahn, "Computing the feasible spaces of optimal power flow problems," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4752–4763, 2017.
- [7] W. A. Bukhsh, A. Grothey, K. I. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, 2013.
- [8] C. Barrows, S. Blumsack, and P. Hines, "Correcting optimal transmission switching for AC power flows," in *47th Hawaii Int. Conf. Syst. Sci.*, January 2014, pp. 2374–2379.
- [9] W. van Ackooij, I. Danti Lopez, A. Frangioni, F. Lacalandra, and M. Tahanaan, "Large-scale unit commitment under uncertainty: An updated literature survey," *Annals Oper. Res.*, vol. 271, no. 1, pp. 11–85, December 2018.
- [10] B. Taheri and D. K. Molzahn, "AC-informed DC optimal transmission switching problems via parameter optimization," *arXiv:2411.10528*, 2024.
- [11] N. P. Padhy, "Unit commitment—A bibliographical survey," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1196–1205, May 2004.
- [12] J. Momoh, M. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. I and II," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–111, 1999.
- [13] I. Aravena *et al.*, "Recent developments in security-constrained AC optimal power flow: Overview of Challenge 1 in the ARPA-E Grid Optimization Competition," *Oper. Res.*, vol. 71, no. 6, pp. 1977–2014, November–December 2023.
- [14] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," *Proc. IEEE*, vol. 108, no. 9, pp. 1656–1676, 2020.
- [15] 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.
- [16] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.
- [17] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS J. Comput.*, vol. 26, no. 4, pp. 718–734, 2014.
- [18] C. W. Taylor, *Power System Voltage Stability*. McGraw-Hill New York, 1994, vol. 3.
- [19] F. Milano, F. Díaz-González, and J. L. Rueda, "An overview on reactive power control in smart grids," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 347–360, 2013.
- [20] K. Dvijotham and D. K. Molzahn, "Error bounds on the DC power flow approximations: A convex relaxation approach," *55th IEEE Conf. Decis. Control (CDC)*, December 2016.
- [21] K. Baker, "Solutions of DC OPF are never AC feasible," in *12th ACM Int. Conf. Future Energy Syst. (e-Energy)*, 2021, pp. 264–268.
- [22] A. Khodaei, M. Shahidehpour, and S. Kamalinia, "Transmission switching in expansion planning," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1722–1733, 2010.
- [23] C. Coffrin and P. V. Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS J. Comput.*, vol. 26, no. 4, pp. 718–734, 2014.
- [24] 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, 2019.
- [25] B. Taheri, R. K. Gupta, and D. K. Molzahn, "Optimizing parameters of the LinDistFlow power flow approximation for distribution systems," *arXiv:2404.05125*, 2024.
- [26] B. Taheri and D. K. Molzahn, "Optimizing parameters of the DC power flow," *Elect. Power Syst. Res.*, vol. 235, p. 110719, 2024, presented at the 23rd Power Syst. Comput. Conf. (PSCC).
- [27] P. Donti, A. Agarwal, N. V. Bedmutha, L. Pileggi, and J. Z. Kolter, "Adversarially robust learning for security-constrained optimal power flow," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 28 677–28 689.
- [28] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *Int. Conf. Machine Learning (ICML)*, 2017, pp. 136–145.
- [29] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, "Differentiable convex optimization layers," in *Adv. Neural Information Processing Syst. 32 (NeurIPS)*, 2019.
- [30] S. Gopinath and H. L. Hijazi, "Benchmarking large-scale ACOPF solutions and optimality bounds," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2022.
- [31] B. Knueven, J. Ostrowski, and J.-P. Watson, "On mixed-integer programming formulations for the unit commitment problem," *INFORMS J. Comput.*, vol. 32, no. 4, pp. 857–876, 2020.
- [32] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1346–1355, 2008.
- [33] K. W. Hedman, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Optimal transmission switching—Sensitivity analysis and extensions," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1469–1479, 2008.
- [34] C. Guo, H. Nagarajan, and M. Bodur, "Tightening quadratic convex relaxations for the AC optimal transmission switching problem," *arXiv:2212.12097*, 2022.
- [35] Y. Liu and C. Singh, "Reliability evaluation of composite power systems using Markov cut-set method," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 777–785, 2010.
- [36] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, 1990.
- [37] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *Int. Conf. Machine Learning (ICML)*, 2017, pp. 136–145.
- [38] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, "Differentiable convex optimization layers," in *Adv. Neural Information Processing Syst. 32 (NeurIPS)*, 2019.
- [39] S. G. Nash, "A survey of truncated-Newton methods," *J. Computat. Appl. Math.*, vol. 124, no. 1–2, pp. 45–59, 2000.
- [40] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer Science & Business Media, 2006.
- [41] T. Kleinert, M. Labb  , I. Ljubi  , and M. Schmidt, "A survey on mixed-integer programming techniques in bilevel optimization," *EURO J. Comput. Optimiz.*, vol. 9, p. 100007, 2021.
- [42] Y. Kim, S. Leyffer, and T. Munson, "MPEC methods for bilevel optimization problems," in *Bilevel Optimization: Advances and Next Challenges*. Springer, 2020, pp. 335–360.
- [43] IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, "The Power Grid Library for benchmarking AC optimal power flow algorithms," August 2019, *arXiv:1908.02788*.
- [44] 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.