Dual Conic Proxies for AC Optimal Power Flow

Guancheng Qiu, Mathieu Tanneau, Pascal Van Hentenryck
H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, USA
qgc@gatech.edu, {mathieu.tanneau, pascal.vanhentenryck}@isye.gatech.edu

***Abstract*—In recent years, there has been significant interest in the development of machine learning-based optimization proxies for AC Optimal Power Flow (AC-OPF). Although significant progress has been achieved in predicting high-quality primal solutions, no existing learning-based approach can provide valid dual bounds for AC-OPF. This paper addresses this gap by training optimization proxies for a convex relaxation of AC-OPF. Namely, the paper considers a second-order cone (SOC) relaxation of AC-OPF, and proposes a novel architecture that embeds a fast, differentiable (dual) feasibility recovery, thus providing valid dual bounds. The paper combines this new architecture with a self-supervised learning scheme, which alleviates the need for costly training data generation. Extensive numerical experiments on medium- and large-scale power grids demonstrate the efficiency and scalability of the proposed methodology.**

***Index Terms*—AC optimal power flow, convex relaxation, neural network, self-supervised learning**

I. Introduction

The AC Optimal Power Flow (AC-OPF) problem determines the most economical generation dispatch to meet current demand while satisfying physical and engineering constraints. Despite being fundamental to power systems operations, AC-OPF is seldom used in practice due to its nonlinear and non-convex nature, which makes it challenging to solve. Instead, most practitioners and market operators rely on the DC-OPF approximation. The computational limitations of AC-OPF, combined with the fact that similar instances need to be solved repeatedly, has spurred significant interest in developing optimization proxies for AC-OPF, i.e., machine learning (ML) models that approximate the input-output mapping of an AC-OPF solver. Once trained, AC-OPF proxies can produce high-quality solutions in milliseconds.

Virtually all existing works on optimization proxies for DC- and AC-OPF focus on predicting primal solutions, i.e., generation dispatches, voltage setpoints, and power flows. Several approaches seek to improve the feasibility of predicted solutions by embedding constraints into the architecture and/or loss function [1]–[4]. Other approaches use feasibility restoration techniques, either as a post-processing step [5], [6] or as a differentiable restoration layer incorporated in an end-to-end manner [2], [4]. Restoring feasibility of a close-to-feasible solution is typically significantly faster than solving the original problem.

This research is partly funded by NSF award 2112533 and ARPA-E PERFORM award AR0001136. The authors would like to thank the anonymous reviewers, whose valuable feedback helped improve this paper.

Despite the impressive progress in developing optimization proxies for AC-OPF, a major limitation remains: *existing optimization proxies do not provide certificates of (near-)optimality.* This contrasts with global optimization solvers, which provide both a primal solution and a certificate of (near-)optimality, i.e., a valid dual bound. The absence of optimality guarantees prevents the deployment of optimization proxies in real systems, because sub-optimal solutions may result in significant economic shortfalls, and electricity markets typically require (near-)optimal solutions [7]. Recent works on neural network verification [8], [9] do offer worst-case performance guarantees for trained optimization proxies. Nevertheless, this approach is restricted to specific classes of models, does not scale well to large systems, and only provides a worst-case guarantee which may not be informative in practice.

The paper addresses this fundamental gap by learning *dual* optimization proxies that provide valid dual bounds in milliseconds. This is the first learning-based approach to provide valid dual bounds for AC-OPF, which is valuable in its own right. The paper makes the following contributions.

1) It proposes dual optimization proxies that leverage convex relaxations of AC-OPF to obtain *valid* dual bounds in milliseconds.
2) It develops a novel dual-feasible architecture, *Dual Conic Proxies* (DCP), that uses conic duality to guarantee dual feasibility.
3) It employs a self-supervised training algorithm to train DCP, thus eliminating the need for costly, and potentially numerically unstable, optimization solvers.
4) It conducts numerical experiments on large-scale power systems, which demonstrate that the DCP approach is scalable and provides high-quality certified dual bounds compared to a conic optimization solver.

The rest of the paper is organized as follows. Section II surveys the relevant literature. Section III presents the AC-OPF problem, its second order cone (SOC) relaxation, and the dual of the SOC relaxation. Section IV describes the dual-feasible architecture and self-supervised training. Section V reports numerical results, and Section VI concludes the paper.

II. Related Work

A. Convex Relaxations of AC-OPF

Convex relaxations of AC-OPF have attracted significant interest in the literature [10], [11]. They include, among others, second-order cone (SOC) [12], semidefinite programming (SDP) [13], quadratic convex (QC) [14], and moment-based

relaxations [15]. While the AC-OPF problem is non-convex and NP-hard [16], convex relaxations can be solved efficiently, i.e., in polynomial time, thus providing certificates of (near-)optimality for AC-OPF solutions [17], [18].

While convex relaxations have received a lot of attention from the optimization and power systems communities, they have seldom been considered in ML settings. Infeasibility certificates from convex relaxations are used in [19], [20] to accelerate the generation of datasets of AC-OPF instances. A classifier model is trained in [21] to predict which variables should be considered in an optimality-based bound tightening (OBBT) scheme, which results in improved computational performance. To the best of the authors' knowledge, the use of ML tools to predict solutions of convex relaxations of AC-OPF has not been considered before. *This paper is the first to propose ML-based dual conic proxies that are guaranteed to output dual-feasible solutions, thus providing valid lower bounds on the optimal value of AC-OPF.*

B. Optimization Proxies for AC-OPF

The development of optimization proxies for OPF problems has seen a surge of interest in recent years, mostly centered around DC-OPF [4], [8], [22]–[26] and AC-OPF [1], [5], [27]–[30]. An overview of these efforts is presented next; an exhaustive review of optimization proxies for OPF problems is beyond the scope of this paper.

Virtually all existing approaches on optimization proxies for OPF problems focus on predicting primal solutions, namely, generation dispatches, voltage setpoints, and power flows. Existing approaches have been applied to generating solutions for DC-OPF [8], [22]–[24] and AC-OPF [1], [5], [27], [28], [30]. In particular, several techniques have been proposed to improve the feasibility of predicted solutions. Physics-informed architectures penalize constraint violations in the loss function [1], [2], [27], [28], [30], [31]. Equality completion is used in [2], [5], [28] to ensure the predicted solution satisfy power flow equations; this step requires solving a system of linear or nonlinear equations. A post-processing feasibility restoration step is also employed in [6], [23]. Finally, a few papers [3], [29]–[31] jointly predict primal and dual solutions. [8], [31] penalize the violation of KKT conditions during training, [3] consider an Augmented Lagrangian-based training scheme, and [29], [30] feeds the predicted primal-dual solution as warm-start to an AC-OPF solver. With the exception of [4], [25], [26] which ensure feasibility of output DC-OPF solutions, the above approaches do not offer feasibility guarantees, unless a post-processing restoration step is employed.

Traditionally, optimization proxies are trained using supervised learning, wherein one first generates a dataset of OPF instances and their solutions, then trains a model to minimize the distance between predicted and ground truth solutions. More recently, self-supervised approaches [2]–[4], [32], [33] directly minimize the objective value of the predicted solution, thereby eliminating the need for costly data generation. As pointed out in [4], a key challenge in self-supervised approaches is to ensure feasibility to avoid predicting trivial infeasible solutions.

TABLE I
THE NOMENCLATURE OF SETS.

Symbol	Description
$\mathcal{N}$	the set of buses, each represented as a node i
$\mathcal{E}$	the set of oriented branches, each represented as a directed arc (i,j)
$\mathcal{E}_i^+$	the set of arcs leaving bus i
$\mathcal{E}_i^-$	the set of arcs entering bus i
$\mathcal{Q}^n$	$\{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + ... + x_n^2}\}$
$\mathcal{Q}_r^n$	$\{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + ... + x_n^2,\ x_1, x_2 \geq 0\}$

C. Optimization Proxy Verification

Due to the risks involved in power system operations, neural network verification has been used to provide guarantees on output quality of OPF optimization proxies that are based on deep neural networks (DNN). Mixed-integer-linear-programming (MILP)-based verification is used in [8], [9] to provide guarantees of worst-case violation and worst-case sub-optimality of a DC-OPF proxy. The sub-optimality verification requires solving a bi-level problem, which is computationally expensive for large systems. [31] uses a mixed integer quadratic program (MIQP) to verify the worst-case constraint violation of a physics-informed neural network for AC-OPF. The work does not consider verification of optimality. [34] develops a more tractable MIQP-based verification to verify the worst-case constraint violation for AC-OPF proxies, but does not consider the quality of the proposed approximation. [35] accelerates the verification process for constraint violations (but not verification of optimality) by adapting it to a specialized, parallelized verification solver. The above methods are based on an approach which relies on an MILP encoding of the neural networks, so they only support models that are MILP-representable.

III. AC-OPF AND A SECOND-ORDER CONE RELAXATION

Throughout the paper, the imaginary unit is denoted by $\mathbf{j}$, i.e., $\mathbf{j}^2 = -1$. The complex conjugate of $z \in \mathbb{C}$ is $z^\star$. Additional notations used by the paper are defined in Tables I and II. For ease of reading and without loss of generality, the presentation assumes that exactly one generator is connected to each bus, and that all costs are linear.

A. The AC Optimal Power Flow Formulation

The formulation of AC-OPF considered in this paper is detailed in Model 1. The objective (1a) minimizes total generation costs. Constraint (1b) enforces power balance (Kirchhoff's node law) at each bus. Constraints (1c)–(1d) enforce Ohm's law for forward and reverse power flows. Constraints (1e) enforce thermal limits on each branch. Finally, constraints (1f)–(1i) enforce minimum and maximum limits on nodal voltage magnitude, phase angle differences, and active/reactive generation.

TABLE II
THE NOMENCLATURE OF VARIABLES AND PARAMETERS IN FORMULATIONS.

Symbol	Description
$\mathbf{S}_i^{\mathrm{g}} = \mathbf{p}_i^{\mathrm{g}} + \mathbf{j}\mathbf{q}_i^{\mathrm{g}}$	power generation at bus i
$\mathbf{V}_i = \mathbf{v}_i \angle \theta_i$	voltage at bus i
$\vec{\mathbf{S}}_{ij} = \vec{\mathbf{p}}_{ij} + \mathbf{j}\vec{\mathbf{q}}_{ij}$ $\overleftarrow{\mathbf{S}}_{ij} = \overleftarrow{\mathbf{p}}_{ij} + \mathbf{j}\overleftarrow{\mathbf{q}}_{ij}$	power flow from bus i to bus j on branch (i,j) power flow from bus j to bus i on branch (i,j)
c_i	linear cost coefficient of power generation at bus i
$\mathrm{S}_i^{\mathrm{d}} = \mathrm{p}_i^{\mathrm{d}} + \mathbf{j}\mathrm{q}_i^{\mathrm{d}}$	power demand at bus i
Y_{ij} Y_{ij}^c	complex line admittance of branch (i,j) complex shunt admittance of branch (i,j)
Y_i^s	complex admittance of shunt at bus i
T_{ij}	complex transformer tap ratio of branch (i,j)
$\bar{s}_{ij}$	thermal limit of branch (i,j)
$\underline{\mathrm{p}}_i^{\mathrm{g}}$, $\overline{\mathrm{p}}_i^{\mathrm{g}}$	bounds on active power generation at bus i
$\underline{\mathrm{q}}_i^{\mathrm{g}}$, $\overline{\mathrm{q}}_i^{\mathrm{g}}$	bounds on reactive power generation at bus i
$\underline{\mathrm{v}}_i$, $\overline{\mathrm{v}}_i$	bounds on voltage magnitude at bus i
$\underline{\theta}_{ij}$, $\overline{\theta}_{ij}$	bounds on angle difference of bus i and bus j

Model 1 The AC-OPF model

$$\min \quad \sum_{i\in\mathcal{N}} c_i \mathbf{p}_i^{\mathrm{g}} \tag{1a}$$

$$\text{s.t.} \quad \mathbf{S}_i^{\mathrm{g}} - \sum_{e\in\mathcal{E}_i^+} \vec{\mathbf{S}}_e - \sum_{e\in\mathcal{E}_i^-} \overleftarrow{\mathbf{S}}_e - (Y_i^s)^\star |\mathbf{V}_i|^2 = \mathrm{S}_i^{\mathrm{d}} \quad \forall i \in \mathcal{N} \tag{1b}$$

$$\vec{\mathbf{S}}_{ij} = (Y_{ij} + Y_{ij}^c)^\star \frac{|\mathbf{V}_i|^2}{|T_{ij}|^2} - Y_{ij}^\star \frac{\mathbf{V}_i \mathbf{V}_j^\star}{T_{ij}} \quad \forall ij \in \mathcal{E} \tag{1c}$$

$$\overleftarrow{\mathbf{S}}_{ij} = (Y_{ij} + Y_{ji}^c)^\star |\mathbf{V}_j|^2 - Y_{ij}^\star \frac{\mathbf{V}_i^\star \mathbf{V}_j}{T_{ij}^\star} \quad \forall ij \in \mathcal{E} \tag{1d}$$

$$|\vec{\mathbf{S}}_{ij}|, |\overleftarrow{\mathbf{S}}_{ij}| \le \bar{s}_{ij} \quad \forall ij \in \mathcal{E} \tag{1e}$$

$$\underline{\mathrm{v}}_i \le |\mathbf{V}_i| \le \overline{\mathrm{v}}_i \quad \forall i \in \mathcal{N} \tag{1f}$$

$$\underline{\theta}_{ij} \le \theta_i - \theta_j \le \overline{\theta}_{ij} \quad \forall ij \in \mathcal{E} \tag{1g}$$

$$\underline{\mathrm{p}}_i^{\mathrm{g}} \le \mathbf{p}_i^{\mathrm{g}} \le \overline{\mathrm{p}}_i^{\mathrm{g}} \quad \forall i \in \mathcal{N} \tag{1h}$$

$$\underline{\mathrm{q}}_i^{\mathrm{g}} \le \mathbf{q}_i^{\mathrm{g}} \le \overline{\mathrm{q}}_i^{\mathrm{g}} \quad \forall i \in \mathcal{N} \tag{1i}$$

B. The Second-Order Cone Relaxation of the AC-OPF

The second-order cone (SOC) relaxation of AC-OPF proposed by Jabr in [12] is obtained by introducing changes of variables

$$\mathbf{w}_i = \mathbf{v}_i^2 \qquad \forall i \in \mathcal{N} \tag{2}$$

$$\mathbf{w}_{ij}^{\mathrm{re}} = \mathbf{v}_i \mathbf{v}_j \cos(\theta_i - \theta_j) \qquad \forall ij \in \mathcal{E} \tag{3}$$

$$\mathbf{w}_{ij}^{\mathrm{im}} = \mathbf{v}_i \mathbf{v}_j \sin(\theta_i - \theta_j) \qquad \forall ij \in \mathcal{E} \tag{4}$$

together with the additional non-convex constraints

$$(\mathbf{w}_{ij}^{\mathrm{re}})^2 + (\mathbf{w}_{ij}^{\mathrm{im}})^2 = \mathbf{w}_i \mathbf{w}_j \qquad \forall ij \in \mathcal{E}. \tag{5}$$

The non-convex quadratic constraints (5) are then relaxed as

$$(\mathbf{w}_{ij}^{\mathrm{re}})^2 + (\mathbf{w}_{ij}^{\mathrm{im}})^2 \le \mathbf{w}_i \mathbf{w}_j \qquad \forall ij \in \mathcal{E}. \tag{6}$$

The resulting SOC-OPF formulation (in real notations) is presented in Model 2, with each constraint's dual variable indicated in brackets. Constraints (7b)–(7c) enforce active and reactive power balance at each bus. Constraints (7d)–(7g) formulate Ohm's law for active/reactive forward and reverse power flows, where constants $\vec{\gamma}, \overleftarrow{\gamma}$ are obtained from Eq. (1c)–(1d). Conic constraints (7h)–(7i) enforce thermal limits on each branch's forward and reverse power flow. Constraints (7j) ensure phase angle difference limits on each branch, and constraint (7k) is Jabr's inequality (6) in conic form. Finally, constraints (7l)–(7n) enforce minimum and maximum limits on nodal voltage magnitude and active/reactive dispatch.

Model 2 The SOC-OPF Model.

$$\min \quad \sum_{i\in\mathcal{N}} c_i \mathbf{p}_i^{\mathrm{g}} \tag{7a}$$

$$\text{s.t.} \quad \mathbf{p}_i^{\mathrm{g}} - \mathrm{p}_i^{\mathrm{d}} - g_i^s \mathbf{w}_i = \sum_{e\in\mathcal{E}_i^+} \vec{\mathbf{p}}_e - \sum_{e\in\mathcal{E}_i^-} \overleftarrow{\mathbf{p}}_e \quad \forall i \in \mathcal{N} \quad [\lambda_i^{\mathrm{p}}] \tag{7b}$$

$$\mathbf{q}_i^{\mathrm{g}} - \mathrm{q}_i^{\mathrm{d}} + b_i^s \mathbf{w}_i = \sum_{e\in\mathcal{E}_i^+} \vec{\mathbf{q}}_e - \sum_{e\in\mathcal{E}_i^-} \overleftarrow{\mathbf{q}}_e \quad \forall i \in \mathcal{N} \quad [\lambda_i^{\mathrm{q}}] \tag{7c}$$

$$\vec{\mathbf{p}}_{ij} = \vec{\gamma}_{ij}^{\mathrm{p,w}} \mathbf{w}_i + \vec{\gamma}_{ij}^{\mathrm{p,r}} \mathbf{w}_{ij}^{\mathrm{re}} + \vec{\gamma}_{ij}^{\mathrm{p,i}} \mathbf{w}_{ij}^{\mathrm{im}} \quad \forall ij \in \mathcal{E} \quad [\vec{\lambda}_{ij}^{\mathrm{p}}] \tag{7d}$$

$$\overleftarrow{\mathbf{p}}_{ij} = \overleftarrow{\gamma}_{ij}^{\mathrm{p,w}} \mathbf{w}_j + \overleftarrow{\gamma}_{ij}^{\mathrm{p,r}} \mathbf{w}_{ij}^{\mathrm{re}} + \overleftarrow{\gamma}_{ij}^{\mathrm{p,i}} \mathbf{w}_{ij}^{\mathrm{im}} \quad \forall ij \in \mathcal{E} \quad [\overleftarrow{\lambda}_{ij}^{\mathrm{p}}] \tag{7e}$$

$$\vec{\mathbf{q}}_{ij} = \vec{\gamma}_{ij}^{\mathrm{q,w}} \mathbf{w}_i + \vec{\gamma}_{ij}^{\mathrm{q,r}} \mathbf{w}_{ij}^{\mathrm{re}} + \vec{\gamma}_{ij}^{\mathrm{q,i}} \mathbf{w}_{ij}^{\mathrm{im}} \quad \forall ij \in \mathcal{E} \quad [\vec{\lambda}_{ij}^{\mathrm{q}}] \tag{7f}$$

$$\overleftarrow{\mathbf{q}}_{ij} = \overleftarrow{\gamma}_{ij}^{\mathrm{q,w}} \mathbf{w}_j + \overleftarrow{\gamma}_{ij}^{\mathrm{q,r}} \mathbf{w}_{ij}^{\mathrm{re}} + \overleftarrow{\gamma}_{ij}^{\mathrm{q,i}} \mathbf{w}_{ij}^{\mathrm{im}} \quad \forall ij \in \mathcal{E} \quad [\overleftarrow{\lambda}_{ij}^{\mathrm{q}}] \tag{7g}$$

$$(\bar{s}_{ij}, \vec{\mathbf{p}}_{ij}, \vec{\mathbf{q}}_{ij}) \in \mathcal{Q}^3 \quad \forall ij \in \mathcal{E} \quad [\vec{\nu}_{ij}] \tag{7h}$$

$$(\bar{s}_{ij}, \overleftarrow{\mathbf{p}}_{ij}, \overleftarrow{\mathbf{q}}_{ij}) \in \mathcal{Q}^3 \quad \forall ij \in \mathcal{E} \quad [\overleftarrow{\nu}_{ij}] \tag{7i}$$

$$\tan(\underline{\theta}_{ij}) \mathbf{w}_{ij}^{\mathrm{re}} \le \mathbf{w}_{ij}^{\mathrm{im}} \le \tan(\overline{\theta}_{ij}) \mathbf{w}_{ij}^{\mathrm{re}} \quad \forall ij \in \mathcal{E} \quad [\mu_{ij}^{\theta}] \tag{7j}$$

$$(\frac{\mathbf{w}_i}{\sqrt{2}}, \frac{\mathbf{w}_j}{\sqrt{2}}, \mathbf{w}_{ij}^{\mathrm{re}}, \mathbf{w}_{ij}^{\mathrm{im}}) \in \mathcal{Q}_r^4 \quad \forall ij \in \mathcal{E} \quad [\omega_{ij}] \tag{7k}$$

$$\underline{\mathrm{v}}_i^2 \le \mathbf{w}_i \le \overline{\mathrm{v}}_i^2 \quad \forall i \in \mathcal{N} \quad [\mu_i^{\mathrm{w}}] \tag{7l}$$

$$\underline{\mathrm{p}}_i^{\mathrm{g}} \le \mathbf{p}_i^{\mathrm{g}} \le \overline{\mathrm{p}}_i^{\mathrm{g}} \quad \forall i \in \mathcal{N} \quad [\mu_i^{\mathrm{pg}}] \tag{7m}$$

$$\underline{\mathrm{q}}_i^{\mathrm{g}} \le \mathbf{q}_i^{\mathrm{g}} \le \overline{\mathrm{q}}_i^{\mathrm{g}} \quad \forall i \in \mathcal{N} \quad [\mu_i^{\mathrm{qg}}] \tag{7n}$$

This paper focuses on the dual problem DSOC-OPF, formulated in Model 3. By convention, dual variables corresponding to two-sided linear inequality constraints are denoted by $\underline{\mu}$ and $\bar{\mu}$, respectively. For instance, $\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}$ denote the dual variables associated to active power generation lower and upper bounds. Also note that the conic dual variables $\vec{\nu}, \overleftarrow{\nu}, \omega$ associated to constraints (7h), (7i), (7k) are vectors, as stated in constraints (8l), (8m). It is important to note that the DSOC-OPF is a second-order conic optimization problem and that, by weak duality, any feasible solution for the DSOC-OPF yields a valid lower bound on the objective value of the SOC-OPF and, in turn, of the AC-OPF. Dual problems, such as the DSOC-OPF, have received very little attention in the literature; to the best of the author's knowledge, *this paper is the first to apply machine learning techniques to the DSOC-OPF*.

IV. THE DUAL-FEASIBLE PROXY ARCHITECTURE

This section presents the proposed DCP architecture, illustrated in Figure 1. DCP predicts a subset of variables (the *independent* variables) before applying a completion step that computes values of the remaining variables (the *dependent* variables) that yields a dual-feasible solution. At a high level

Model 3 The DSOC-OPF Model.

$$\max_{\lambda,\mu,\nu} \quad \sum_{i\in\mathcal{N}} \left(\mathrm{p}_i^{\mathrm{d}}\lambda_i^{\mathrm{p}} + \mathrm{q}_i^{\mathrm{d}}\lambda_i^{\mathrm{q}} + \underline{\mathrm{p}}_i^{\mathrm{g}}\underline{\mu}_i^{\mathrm{pg}} - \overline{\mathrm{p}}_i^{\mathrm{g}}\bar{\mu}_i^{\mathrm{pg}} + \underline{\mathrm{q}}_i^{\mathrm{g}}\underline{\mu}_i^{\mathrm{qg}} - \overline{\mathrm{q}}_i^{\mathrm{g}}\bar{\mu}_i^{\mathrm{qg}} + \underline{\mathrm{v}}_i^2\underline{\mu}_i^{\mathrm{w}} - \overline{\mathrm{v}}_i^2\bar{\mu}_i^{\mathrm{w}}\right) - \sum_{e\in\mathcal{E}} \bar{s}_e\,(\vec{\nu}_e^{\,\mathrm{s}} + \overleftarrow{\nu}_e^{\,\mathrm{s}}) \tag{8a}$$

$$s.t. \quad \lambda_i^{\mathrm{p}} + \underline{\mu}_i^{\mathrm{pg}} - \bar{\mu}_i^{\mathrm{pg}} = c_i \qquad \forall i\in\mathcal{N} \tag{8b}$$

$$\lambda_i^{\mathrm{q}} + \underline{\mu}_i^{\mathrm{qg}} - \bar{\mu}_i^{\mathrm{qg}} = 0 \qquad \forall i\in\mathcal{N} \tag{8c}$$

$$-\lambda_i^{\mathrm{p}} - \vec{\lambda}_{ij}^{\mathrm{p}} + \vec{\nu}_{ij}^{\mathrm{p}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8d}$$

$$-\lambda_i^{\mathrm{q}} - \vec{\lambda}_{ij}^{\mathrm{q}} + \vec{\nu}_{ij}^{\mathrm{q}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8e}$$

$$-\lambda_j^{\mathrm{p}} - \overleftarrow{\lambda}_{ij}^{\mathrm{p}} + \overleftarrow{\nu}_{ij}^{\mathrm{p}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8f}$$

$$-\lambda_j^{\mathrm{q}} - \overleftarrow{\lambda}_{ij}^{\mathrm{q}} + \overleftarrow{\nu}_{ij}^{\mathrm{q}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8g}$$

$$-g_i^{\mathrm{s}}\lambda_i^{\mathrm{p}} + b_i^{\mathrm{s}}\lambda_i^{\mathrm{q}} + \sum_{e\in\mathcal{E}_i^+}\left(\vec{\gamma}_e^{\mathrm{p,w}}\vec{\lambda}_e^{\mathrm{p}} + \vec{\gamma}_e^{\mathrm{q,w}}\vec{\lambda}_e^{\mathrm{q}} + \frac{\omega_e^{\mathrm{f}}}{\sqrt{2}}\right) + \sum_{e\in\mathcal{E}_i^-}\left(\overleftarrow{\gamma}_e^{\mathrm{p,w}}\overleftarrow{\lambda}_e^{\mathrm{p}} + \overleftarrow{\gamma}_e^{\mathrm{q,w}}\overleftarrow{\lambda}_e^{\mathrm{q}} + \frac{\omega_e^{\mathrm{t}}}{\sqrt{2}}\right) + \underline{\mu}_i^{\mathrm{w}} - \bar{\mu}_i^{\mathrm{w}} = 0 \qquad \forall i\in\mathcal{N} \tag{8h}$$

$$\vec{\gamma}_{ij}^{\mathrm{p,r}}\vec{\lambda}_{ij}^{\mathrm{p}} + \overleftarrow{\gamma}_{ij}^{\mathrm{p,r}}\overleftarrow{\lambda}_{ij}^{\mathrm{p}} + \vec{\gamma}_{ij}^{\mathrm{q,r}}\vec{\lambda}_{ij}^{\mathrm{q}} + \overleftarrow{\gamma}_{ij}^{\mathrm{q,r}}\overleftarrow{\lambda}_{ij}^{\mathrm{q}} - \tan(\underline{\theta}_{ij})\underline{\mu}_{ij}^{\theta} + \tan(\bar{\theta}_{ij})\bar{\mu}_{ij}^{\theta} + \omega_{ij}^{\mathrm{re}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8i}$$

$$\vec{\gamma}_{ij}^{\mathrm{p,i}}\vec{\lambda}_{ij}^{\mathrm{p}} + \overleftarrow{\gamma}_{ij}^{\mathrm{p,i}}\overleftarrow{\lambda}_{ij}^{\mathrm{p}} + \vec{\gamma}_{ij}^{\mathrm{q,i}}\vec{\lambda}_{ij}^{\mathrm{q}} + \overleftarrow{\gamma}_{ij}^{\mathrm{q,i}}\overleftarrow{\lambda}_{ij}^{\mathrm{q}} + \underline{\mu}_{ij}^{\theta} - \bar{\mu}_{ij}^{\theta} + \omega_{ij}^{\mathrm{im}} = 0 \qquad \forall ij\in\mathcal{E} \tag{8j}$$

$$\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}, \underline{\mu}^{\mathrm{qg}}, \bar{\mu}^{\mathrm{qg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}, \underline{\mu}^{\theta}, \bar{\mu}^{\theta} \geq 0 \tag{8k}$$

$$\vec{\nu}_{ij} = (\vec{\nu}_{ij}^{\mathrm{s}}, \vec{\nu}_{ij}^{\mathrm{p}}, \vec{\nu}_{ij}^{\mathrm{q}}) \in \mathcal{Q}^3, \overleftarrow{\nu}_{ij} = (\overleftarrow{\nu}_{ij}^{\mathrm{s}}, \overleftarrow{\nu}_{ij}^{\mathrm{p}}, \overleftarrow{\nu}_{ij}^{\mathrm{q}}) \in \mathcal{Q}^3 \qquad \forall ij\in\mathcal{E} \tag{8l}$$

$$\omega_{ij} = \left(\omega_{ij}^{\mathrm{f}}, \omega_{ij}^{\mathrm{t}}, \omega_{ij}^{\mathrm{re}}, \omega_{ij}^{\mathrm{im}}\right) \in \mathcal{Q}_r^4, \qquad \forall ij\in\mathcal{E} \tag{8m}$$

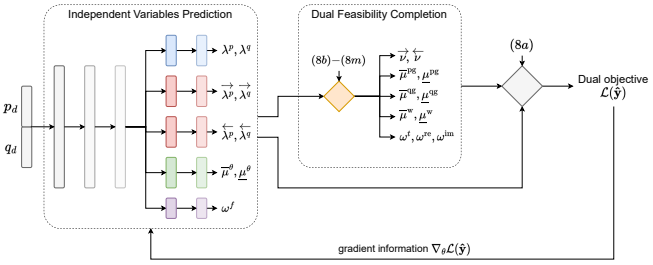

Fig. 1. The Proposed Dual Conic Proxy Architecture.

(and with a slight approximation), the prediction step of DCP delivers dual values for physical and phase angle constraints, while its completion step produces values for the remaining duals. *The completion step of DCP is a key contribution of the paper. Contrary to existing completion approaches, e.g., [2], that only use equality constraints, DCP leverages fundamental properties of dual-optimal solutions.*

In the following, $\mathbf{x}$ (resp. $\mathbf{y}$) denotes the vector of all primal (resp. dual) variables. It is easy to verify that SOC-OPF is always bounded, and that DSOC-OPF is always strictly feasible. In addition, the paper assumes that SOC-OPF is strictly feasible. Therefore, by strong conic duality (see Theorem 1.4.2 in [36]), both problems are solvable, and any primal-dual optimal solution satisfies complementary slackness.

A. Dual Feasibility Completion

The dual-feasibility completion takes as inputs a prediction for a set of *independent* variables, and outputs values for the remaining *dependent* variables that ensure that Constraints (8b)–(8m) are satisfied by the resulting assignment of all variables. Existing completion approaches, e.g., [2], only use equality constraints. In contrast, DCP also leverages properties of dual-optimal solutions, which are formalized in the following Lemmas.

Lemma 1. *Let* $\mathbf{y}$ *be dual-optimal. Then,*

$$\forall i\in\mathcal{N},\ \underline{\mu}_i^{pg}\times\bar{\mu}_i^{pg} = \underline{\mu}_i^{qg}\times\bar{\mu}_i^{qg} = \underline{\mu}_i^{w}\times\bar{\mu}_i^{w} = 0.$$

Proof. The proof is given for $\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}$; it is similar for $\underline{\mu}^{\mathrm{qg}}, \bar{\mu}^{\mathrm{qg}}$ and $\underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}$. Let $i\in\mathcal{N}$, and assume $\underline{\mu}_i^{\mathrm{pg}}\times\bar{\mu}_i^{\mathrm{pg}} > 0$, i.e., $\underline{\mu}_i^{\mathrm{pg}}, \bar{\mu}_i^{\mathrm{pg}} > 0$ because of (8k). Define $\delta = \min\{\underline{\mu}_i^{\mathrm{pg}}, \bar{\mu}_i^{\mathrm{pg}}\} > 0$. Then, $(\underline{\mu}_i^{\mathrm{pg}} - \delta, \bar{\mu}_i^{\mathrm{pg}} - \delta)$ yields a dual-feasible solution, and increases the objective by $(\overline{\mathrm{p}}_i^{\mathrm{g}} - \underline{\mathrm{p}}_i^{\mathrm{g}})\delta > 0$, which contradicts the assumption that $\mathbf{y}$ is optimal. □

Lemma 2. *Let* $\mathbf{y}$ *be dual-optimal. Then,*

$$\forall ij\in\mathcal{E},\ \vec{\nu}_{ij}^{\,s} = \sqrt{(\vec{\nu}_{ij}^{\,p})^2 + (\vec{\nu}_{ij}^{\,q})^2},\ \overleftarrow{\nu}_{ij}^{\,s} = \sqrt{(\overleftarrow{\nu}_{ij}^{\,p})^2 + (\overleftarrow{\nu}_{ij}^{\,q})^2}.$$

Proof. Variables $\vec{\nu}_{ij}^{\mathrm{s}}$, $\overleftarrow{\nu}_{ij}^{\mathrm{s}}$ only appear in the SOC constraints (8l), and have negative objective coefficients. □

Lemma 3. *Let* $\mathbf{y}$ *be dual-optimal. Then,*

$$\forall ij\in\mathcal{E},\ 2\omega_i^f\omega_j^t = (\omega_{ij}^{re})^2 + (\omega_{ij}^{im})^2 \tag{9}$$

Proof. The result is a direct consequence of conic complementary slackness which, in the current setting, reads

$$\left(\frac{\mathbf{w}_i}{\sqrt{2}}, \frac{\mathbf{w}_j}{\sqrt{2}}, \mathbf{w}_{ij}^{\mathrm{re}}, \mathbf{w}_{ij}^{\mathrm{im}}\right)^\top \left(\omega_{ij}^{\mathrm{f}}, \omega_{ij}^{\mathrm{t}}, \omega_{ij}^{\mathrm{re}}, \omega_{ij}^{\mathrm{im}}\right) = 0 \tag{10}$$

In practice, $\underline{\mathrm{v}} > 0$, so constraints (7l) impose $\mathbf{w}_i, \mathbf{w}_j > 0$, which yields $\left(\frac{\mathbf{w}_i}{\sqrt{2}}, \frac{\mathbf{w}_j}{\sqrt{2}}, \mathbf{w}_{ij}^{\mathrm{re}}, \mathbf{w}_{ij}^{\mathrm{im}}\right) \neq 0$. So for complementary slackness (10) to hold, $\left(\omega_{ij}^{\mathrm{f}}, \omega_{ij}^{\mathrm{t}}, \omega_{ij}^{\mathrm{re}}, \omega_{ij}^{\mathrm{im}}\right)$ cannot be strictly conic-feasible, i.e., $\left(\omega_{ij}^{\mathrm{f}}, \omega_{ij}^{\mathrm{t}}, \omega_{ij}^{\mathrm{re}}, \omega_{ij}^{\mathrm{im}}\right) \notin \mathrm{int}(\mathcal{Q}_r^4)$. Thus, constraint (8m) is active, i.e., $2\omega_{ij}^{\mathrm{f}}\omega_{ij}^{\mathrm{t}} = (\omega_{ij}^{\mathrm{re}})^2 + (\omega_{ij}^{\mathrm{im}})^2$. □

The DCP framework applied to DSOC-OPF is presented in Algorithm 1. Lemma 1 can be used to recover variables $\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}$, $\underline{\mu}^{\mathrm{qg}}, \bar{\mu}^{\mathrm{qg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}$ using constraints (8b), (8c), and

Algorithm 1: The DCP Methodology for DSOC-OPF.

1 Predict $\lambda^{\mathrm{p}}, \lambda^{\mathrm{q}}, \vec{\lambda}^{\mathrm{p}}, \vec{\lambda}^{\mathrm{q}}, \overleftarrow{\lambda}^{\mathrm{p}}, \overleftarrow{\lambda}^{\mathrm{q}}, \underline{\mu}^{\theta}, \bar{\mu}^{\theta}, \omega^{\mathrm{f}}$
2 Recover $\vec{\nu}^{\mathrm{p}}, \vec{\nu}^{\mathrm{q}}, \overleftarrow{\nu}^{\mathrm{p}}, \overleftarrow{\nu}^{\mathrm{q}}$ using constraints (8d)–(8g)
3 Recover $\vec{\nu}^{\mathrm{s}}, \overleftarrow{\nu}^{\mathrm{s}}$ using Lemma 2
4 Recover $\omega^{\mathrm{re}}, \omega^{\mathrm{im}}$ using constraints (8i)–(8j)
5 Recover ω^{t} from $\omega^{\mathrm{f}}, \omega^{\mathrm{re}}, \omega^{\mathrm{im}}$ using Lemma 3
6 Recover $\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}, \underline{\mu}^{\mathrm{qg}}, \bar{\mu}^{\mathrm{qg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}$ via (8b), (8c), (8h)

(8h). Indeed, it suffices to impose $\underline{\mu}^{\mathrm{pg}}_i = \max\{0, c_i - \lambda^{\mathrm{p}}_i\}$ and $\bar{\mu}^{\mathrm{pg}}_i = \max\{0, \lambda^{\mathrm{p}}_i - c_i\}$; variables $\underline{\mu}^{\mathrm{qg}}, \bar{\mu}^{\mathrm{qg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}$ are recovered in a similar fashion. Lemma 2 can be used to recover $\vec{\nu}^{\mathrm{s}}, \overleftarrow{\nu}^{\mathrm{s}}$ from $\vec{\nu}^{\mathrm{p}}, \vec{\nu}^{\mathrm{q}}, \overleftarrow{\nu}^{\mathrm{p}}, \overleftarrow{\nu}^{\mathrm{q}}$. Lemma 3 further eliminates $|\mathcal{E}|$ degrees of freedom by proving that (8m) is always active at a dual optimum. This makes it possible to first recover $\omega^{\mathrm{re}}, \omega^{\mathrm{im}}$ using constraints (8i)–(8j), then ω^{t} from $\omega^{\mathrm{f}}, \omega^{\mathrm{re}}, \omega^{\mathrm{im}}$.

Denote by $\xi^{\mathrm{ind}} = (\lambda^{\mathrm{p}}, \lambda^{\mathrm{q}}, \vec{\lambda}^{\mathrm{p}}, \vec{\lambda}^{\mathrm{q}}, \overleftarrow{\lambda}^{\mathrm{p}}, \overleftarrow{\lambda}^{\mathrm{q}}, \underline{\mu}^{\theta}, \bar{\mu}^{\theta}, \omega^{\mathrm{f}})$ and by $\xi^{\mathrm{dep}} = (\vec{\nu}, \overleftarrow{\nu}, \omega^{\mathrm{t}}, \omega^{\mathrm{re}}, \omega^{\mathrm{im}}, \underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}})$ the set of independent and dependent variables, respectively. Recall that Algorithm 1 allows to recover dependent variables $\hat{\xi}^{\mathrm{dep}}$ from the predicted independent variables $\hat{\xi}^{\mathrm{ind}}$. An important property of the recovered solution, is that it attains the best possible dual bound among all possible recovered solutions.

Theorem 1. *Let $\hat{\xi}^{ind}$ be fixed, and define $\hat{\xi}^{dep}$ the recovered dependent variables from Algorithm 1. Then*

$$\hat{\xi}^{dep} \in \arg\max_{\xi^{dep}} \left\{ (8a) \;\middle|\; (8b) - (8m),\ \xi^{ind} = \hat{\xi}^{ind} \right\} \qquad (11)$$

Proof. The proof follows the same steps as the proofs of Lemmas 1–3 and outlined in Algorithm 1. Namely, $\vec{\nu}^{\mathrm{p}}, \vec{\nu}^{\mathrm{q}}, \overleftarrow{\nu}^{\mathrm{p}}, \overleftarrow{\nu}^{\mathrm{q}}$ are uniquely obtained from (8d)–(8g), followed by $\vec{\nu}^{\mathrm{s}}, \overleftarrow{\nu}^{\mathrm{s}}$ using the *optimality conditions* of Lemma 2. Then, $\omega^{\mathrm{re}}, \omega^{\mathrm{im}}$ are uniquely obtained from constraints (8i)–(8j), followed by ω^{t} using the optimality conditions of Lemma 3, i.e., that the conic constraint (8m) must be active in any optimal solution of Problem (11). Finally, $\underline{\mu}^{\mathrm{pg}}, \bar{\mu}^{\mathrm{pg}}, \underline{\mu}^{\mathrm{w}}, \bar{\mu}^{\mathrm{w}}$ are recovered using the optimality conditions of Lemma 1. This concludes the proof by showing that the recovered dual solution is optimal for Problem (11). □

Note that the above dual completion procedure is not unique. Indeed, DSOC-OPF contains $8|\mathcal{N}| + 16|\mathcal{E}|$ degrees of freedom (i.e., scalar variables), $3|\mathcal{N}| + 6|\mathcal{E}|$ linear equality constraints, $6|\mathcal{N}| + 2|\mathcal{E}|$ linear inequality constraints, and $3|\mathcal{E}|$ SOC constraints. Each equality constraint eliminates one degree of freedom. The results of Lemmas 1–3 eliminate another $3|\mathcal{N}|$, $2|\mathcal{E}|$, and $|\mathcal{E}|$ degrees of freedom, respectively. Therefore, any valid completion procedure should be given $2|\mathcal{N}| + 7|\mathcal{E}|$ variables, and recover the remaining $6|\mathcal{N}| + 9|\mathcal{E}|$ via equality constraints and Lemmas 1–3. This leaves ample freedom in the choice of dependent and independent variables, which can significantly impact the performance of the resulting dual conic proxy. For instance, an alternative approach, explored in Section V, is to initially predict $\vec{\nu}^{\mathrm{p}}$, $\vec{\nu}^{\mathrm{q}}$, $\overleftarrow{\nu}^{\mathrm{p}}$, $\overleftarrow{\nu}^{\mathrm{q}}$, and recover $\vec{\lambda}^{\mathrm{p}}$, $\vec{\lambda}^{\mathrm{q}}$, $\overleftarrow{\lambda}^{\mathrm{p}}$, $\overleftarrow{\lambda}^{\mathrm{q}}$ at step 2 of Algorithm 1.

B. Rectangular vs Polar Representation

Another modifiable design in a DSOC-OPF proxy is whether to represent complex quantities in rectangular or polar form. Namely, one may represent $z \in \mathbb{C}$ as $z = p + \mathbf{j}q$ where $p, q \in \mathbb{R}$, or as $z = \rho\angle\theta$ where $\rho \geq 0$, $\theta \in \mathbb{R}$. While optimization formulations may require rectangular or polar form to ensure convexity, there is no such restriction when designing ML models, which are typically non-convex. This work has found that representing $(\omega^{\mathrm{f}}, \omega^{\mathrm{t}})$ in polar form substantially improves optimality of the proxy (see Section V). *To the best of the authors' knowledge, the impact of polar vs rectangular representations on the performance of OPF proxies has not been studied before.*

The dual feasibility completion outlined in Algorithm 1 recovers ω^{t} from $\omega^{\mathrm{f}}, \omega^{\mathrm{re}}, \omega^{\mathrm{im}}$ using Eq. (9). However, this approach is numerically unstable if $\omega^{\mathrm{f}} \ll 1$, and precludes $(\omega^{\mathrm{f}}, \omega^{\mathrm{t}}) = (0, 0)$, which may hold in a dual-optimal solution. Therefore, the paper elects to predict both $\omega^{\mathrm{f}}, \omega^{\mathrm{t}}$, then recover $(\tilde{\omega}^{\mathrm{f}}, \tilde{\omega}^{\mathrm{t}}) = (\omega^{\mathrm{f}} + \delta\omega, \omega^{\mathrm{t}} + \delta\omega)$ that satisfies Eq. (9). Thereby, $\delta\omega$ is obtained by solving a second-order equation. This approach represents $(\omega^{\mathrm{f}}, \omega^{\mathrm{t}})$ in rectangular form.

The result of Lemma 3 shows that $(\omega^{\mathrm{f}}, \omega^{\mathrm{t}})$ reside on a hyperbola, which can be represented using polar coordinates, thus suggesting the following change of variable. Let $(\omega^{\mathrm{f}}, \omega^{\mathrm{t}}) = (\rho\cos\phi, \rho\sin\phi)$ where $\rho \geq 0$ and $\phi \in [0, \pi/2]$. Substituting into Eq. (9), this yields

$$\rho = \sqrt{\frac{(\omega^{\mathrm{re}})^2 + (\omega^{\mathrm{im}})^2}{\sin(2\phi)}}. \qquad (12)$$

Preliminary analysis of optimal solutions provided by optimization solver reveals that the distribution of ϕ angles, across multiple instances, has a mean close to $\pi/4$ and very low variance, which makes it easy to learn. This approach also has the advantage of predicting $|\mathcal{E}|$ fewer independent variables.

C. The Dual Conic Proxy Architecture and Training

The DCP architecture, illustrated in Figure 1, is flexible and can be easily adapted to different ML models, different sets of independent variables, or different internal representations as outlined in Sections IV-A and IV-B.

The first component of the DCP model is a Deep Neural Network (DNN) model that takes, as inputs, active and reactive loads $\mathrm{p}^{\mathrm{d}}, \mathrm{q}^{\mathrm{d}}$, and outputs independent variables. The DNN architecture considered in the experiments consists of an initial set of fully-connected layers, followed by sub-networks, each of which outputs a subset of the independent variables. The use of sub-networks makes the overall architecture more scalable, accounting for the large number of independent variables ($2|\mathcal{N}| + 7|\mathcal{E}|$) to predict.

The second component of the DCP model is the dual feasibility completion. This module takes, as inputs, the independent variables, as well as relevant problem data, and applies Algorithm 1 to recover the values of all dependent variables. The completed solution $\hat{\mathbf{y}}$ satisfies dual feasibility by design. The dual feasibility completion uses closed-form

TABLE III
TEST SYSTEMS STATISTICS

System	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	#indep.	$\mathrm{P}^{d}_{\mathrm{ref}}$ †	$[\underline{\mathrm{P}}^{d}, \overline{\mathrm{P}}^{d}]$‡
ieee14	14	20	168	2.6	[1.9, 2.9]
ieee118	118	186	1,538	42.4	[33.4, 51.8]
ieee300	300	411	3,477	235.3	[184.8, 250.3]
pegase1354	1354	1991	16,645	730.6	[581.3, 772.3]
pegase2869	2869	4585	37,833	1324.4	[1053.9, 1529.8]

†Reference total active power load. ‡Range of total load in the dataset.

formulae, which are supported by most ML libraries, and are differentiable almost everywhere. The latter allows for computing gradient information through back-propagation.

The DCP model is trained in a self-supervised way, using the objective function of the optimization problem for the loss function $\mathcal{L}(\hat{\mathbf{y}})$. The loss function is back-propagated through the completion layer back to the weights of the DNN in an end-to-end manner. Self-supervised training alleviates the need for offline data generation, and has been shown to provide higher-quality solutions compared to supervised training [4]. As pointed out in [4], ensuring feasibility is key to the performance of self-supervised training.

V. Experimental Results

A. Data Generation

The paper conducts experiments on several systems from the PGLib library v21.07 [37]. Table III reports, for each system, the number of buses ($|\mathcal{N}|$) and branches ($|\mathcal{E}|$), the number of independent variables (#indep, equals $2|\mathcal{N}|+7|\mathcal{E}|$), the total active power demand ($\mathrm{P}^{d}_{\mathrm{ref}}$, in per-unit) in the reference case from PGLib, and the range of total active power demand in the dataset ($\underline{\mathrm{P}}^{d}, \overline{\mathrm{P}}^{d}$, see details below).

For each system, instances are generated by perturbing loads in the same fashion as [4], [29]. Namely, the active and reactive loads $\mathrm{p}^{\mathrm{d}}, \mathrm{q}^{\mathrm{d}} \in \mathbb{R}^{N}$ are sampled as

$$\mathrm{p}^{\mathrm{d}} = \alpha \cdot \eta \circ \bar{\mathrm{p}}^{\mathrm{d}}, \quad \mathrm{q}^{\mathrm{d}} = \alpha \cdot \eta \circ \bar{\mathrm{q}}^{\mathrm{d}} \tag{13}$$

where $\bar{\mathrm{p}}^{\mathrm{d}}, \bar{\mathrm{q}}^{\mathrm{d}} \in \mathbb{R}^{|\mathcal{N}|}$ are the vectors of nominal active and reactive demand in the PGLib test case, $\alpha \in \mathbb{R}$ is a system-wide scaling factor, $\eta \in \mathbb{R}^{|\mathcal{N}|}$ is a vector of uncorrelated, multiplicative noise, and $\circ$ denotes element-wise product. The system-wide scaling factor α is sampled from a uniform distribution $U[l, u]$ where l=0.8 and u ranges from 1.05 to 1.20 depending on system capacity, and η is sampled from a log-normal distribution LogNormal$(\frac{-\sigma^2}{2}, \sigma^2)$ with $\sigma = 0.05$. The last column of Table III reports the range of total active power demand in the dataset generated.

For each system, a dataset of SOC-OPF instances and their solutions is generated. Each instance is formulated using PowerModels [38] and solved twice: once with Mosek 10 [39], and once with Clarabel v0.7 [40] in quadruple (128-bit) floating-point arithmetic, denoted by Clarabel$_{\text{quad}}$ in what follows. Mosek uses tolerances of 10^{-6}, in line with [41], whereas Clarabel$_{\text{quad}}$ uses tolerances of 10^{-12} to obtain high-precision solutions. The use of extended precision alleviates numerical issues typically encountered by conic interior-point solvers when solving SOC-OPF problems [41], thus allowing to obtain very high-quality solutions, albeit at the cost of increased computing times. In all that follows, the solutions obtained by Clarabel$_{\text{quad}}$ are considered as "ground-truth" primal-dual optima. Both solvers use a single CPU core to facilitate solving all instances in parallel. Gurobi was not used in the experiments because its API does not support querying conic dual variables.

For each system, a total of 65,536 instances are generated. Mosek failed to converge for 1690, 0, 3269, 72 and 0 instances for the ieee14, ieee118, ieee300, pegase1354 and pegase2869 systems, respectively: specifically, JuMP reported the termination status of these instances to be SLOW_PROGRESS, and feasible primal-dual pairs were not found. Clarabel$_{\text{quad}}$ did not report any numerical issue. Out of these instances, Clarabel$_{\text{quad}}$ finds 2087, 0, 3890, 221 and 1 instances to be infeasible for the systems respectively, and finds solutions for all remaining instances. A dataset of 30,000 instances is then created by randomly selecting instances for which a primal-dual pair was found by both solvers. Finally, each dataset is split into training (90%), validation (5%) and testing (5%). The solutions returned by Mosek and Clarabel$_{\text{quad}}$ are only used for evaluating optimality gaps during testing, i.e., they are not used in training or validation.

The DCP proxies are implemented in Python 3.10 using PyTorch 2.0 [42] with the Adam optimizer [43]. All experiments are conducted on Intel Xeon 2.7GHz CPU machines running Linux and equipped with Tesla RTX 6000 GPUs, at the Phoenix cluster [44].

B. Performance Evaluation

Given the numerical challenges of solving SOC-OPF problems, it is reasonable to expect that the dual solution returned by Mosek may not satisfy all (dual) constraints to high accuracy. Therefore, the dual bound returned by the solver may be invalid. Such behavior has been reported in [41], and in [45] in the context of SDP relaxations of AC-OPF.

To measure this effect, Table IV reports, for each system, the maximum constraint violation of Mosek's dual solutions in the testing set. Mosek uses an interior-point algorithm that strictly enforces variable bounds and conic constraints (8k)–(8m); constraint violations therefore occur on dual equality constraints, especially (8h)–(8j). Recall that, in DSOC-OPF, the equality constraints (8d)–(8j) all have zero right-hand side, thus, only absolute violations are meaningful.

The quality of the DCP and Mosek dual solutions is evaluated in terms of optimality gap with respect to the objective value obtained by Clarabel$_{\text{quad}}$. Recall that the DCP solutions are always dual-feasible, hence the associated dual bound is always valid. To ensure a fair comparison, the same dual recovery steps used in the DCP proxies are applied to repair the Mosek solutions. Note that different dual recovery steps

TABLE IV
MAXIMUM CONSTRAINT VIOLATION OF MOSEK DUAL SOLUTIONS.

System	(8b)–(8c)	(8d)–(8g)	(8h)	(8i)	(8j)
ieee14	0.00	0.00	0.02	0.02	0.02
ieee118	0.01	0.00	0.95	0.17	1.23
ieee300	0.08	0.10	516.51	73.74	211.61
pegase1354	0.01	0.01	93.24	75.62	12.16
pegase2869	0.01	0.01	100.48	100.77	25.64

Constraints (8k)–(8m) are always satisfied. All dual violations in per-unit.

may yield different dual bounds for the same Mosek dual solution, The optimality gap is defined as

$$\text{gap} = \frac{z^* - \hat{z}}{|z^*|},$$

where z^* is the dual objective value of the Clarabel$_{\text{quad}}$ solution, and $\hat{z}$ denotes the objective value of a candidate dual solution, obtained from DCP or Mosek (after dual recovery). Finally, the paper reports averages using the geometric mean $\mu(x_1, ..., x_n) = \sqrt[n]{x_1 x_2 ... x_n}$.

C. Dual Conic Proxy Performance

The experiments consider different realizations of the DCP architecture on multiple test case systems, i.e., it considers different sets of independent variables and the use of polar vs. rectangular form as outlined in Section IV-B. Since the magnitude of different types of dual variables varies greatly in scale, the experiments also study the impact of scaling the outputs of different sub-networks by empirically chosen scaling factors that are in powers of 10. Since this scaling is found to improve optimality gaps in most cases, only these results are shown. Due to space limitations, only the following approaches are reported: (1) whether to predict $\vec{\lambda}^{\text{p}}$, $\vec{\lambda}^{\text{q}}$, $\overleftarrow{\lambda}^{\text{p}}$, $\overleftarrow{\lambda}^{\text{q}}$ and recover $\vec{\nu}^{\text{p}}$, $\vec{\nu}^{\text{q}}$, $\overleftarrow{\nu}^{\text{p}}$, $\overleftarrow{\nu}^{\text{q}}$, or vice versa, and (2) whether to predict $(\omega^{\text{t}}, \omega^{\text{f}})$ in rectangular form or polar form.

Table V reports, for each system and DCP architecture, the mean, standard deviation, and maximum optimality gaps achieved by DCP and by Mosek. Recall that all gaps are measured against the optimal value obtained by Clarabel$_{\text{quad}}$. The second and third columns indentify the DCP architecture, namely, whether $\vec{\lambda}, \overleftarrow{\lambda}$ or $\vec{\nu}, \overleftarrow{\nu}$ are selected as independent variables, and whether $(\omega^{\text{f}}, \omega^{\text{t}})$ is represented using rectangular or polar form. All optimality gaps for Mosek are computed from the repaired Mosek solutions, using the same recovery steps as the DCP architecture.

The results in Table V highlight some important points. First, selecting $\vec{\nu}, \overleftarrow{\nu}$ as independent variables always outperforms $\vec{\lambda}, \overleftarrow{\lambda}$. This is potentially because most branches are never congested in any instance, in which case $\vec{\nu} = \overleftarrow{\nu} = 0$, which greatly simplifies the learning task. In addition, the polar representation of $(\omega^{\text{f}}, \omega^{\text{t}})$ also outperforms the rectangular representation. As mentioned in Section IV-B, this may be explained by the fact that the distribution of angles is typically centred around $\pi/4$ and exhibits very small variance, which also simplifies the learning task.

TABLE V
DCP PERFORMANCE RESULTS

			%gap – DCP			%gap – Mosek†		
System	indep.	$(\omega^{\text{f}}, \omega^{\text{t}})$	mean	std	max	mean	std	max
ieee14	$\vec{\lambda}, \overleftarrow{\lambda}$	Rect.	1.17	5.50	25.20	0.00	0.00	0.01
	$\vec{\lambda}, \overleftarrow{\lambda}$	Polar	0.32	5.53	24.79	0.00	0.00	0.00
	$\vec{\nu}, \overleftarrow{\nu}$	Rect.	0.54	5.48	24.68	0.00	0.00	0.01
	$\vec{\nu}, \overleftarrow{\nu}$	Polar	0.05	5.51	24.52	0.00	0.00	0.01
ieee118	$\vec{\lambda}, \overleftarrow{\lambda}$	Rect.	0.76	0.49	2.51	0.00	0.00	0.00
	$\vec{\lambda}, \overleftarrow{\lambda}$	Polar	0.40	0.45	2.07	0.00	0.00	0.00
	$\vec{\nu}, \overleftarrow{\nu}$	Rect.	0.36	0.09	0.98	0.00	0.00	0.00
	$\vec{\nu}, \overleftarrow{\nu}$	Polar	0.08	0.15	0.57	0.00	0.00	0.00
ieee300	$\vec{\lambda}, \overleftarrow{\lambda}$	Rect.	1.47	1.46	14.86	0.00	0.01	0.18
	$\vec{\lambda}, \overleftarrow{\lambda}$	Polar	0.94	1.46	14.45	0.00	0.01	0.18
	$\vec{\nu}, \overleftarrow{\nu}$	Rect.	0.86	1.37	14.11	0.00	0.01	0.18
	$\vec{\nu}, \overleftarrow{\nu}$	Polar	0.58	1.27	13.56	0.00	0.01	0.18
pegase1354	$\vec{\lambda}, \overleftarrow{\lambda}$	Rect.	54.86	5.01	63.28	0.02	0.01	0.06
	$\vec{\lambda}, \overleftarrow{\lambda}$	Polar	54.77	5.02	63.20	0.02	0.01	0.06
	$\vec{\nu}, \overleftarrow{\nu}$	Rect.	2.14	0.23	2.66	0.01	0.00	0.04
	$\vec{\nu}, \overleftarrow{\nu}$	Polar	1.62	0.18	2.08	0.01	0.00	0.04
pegase2869	$\vec{\lambda}, \overleftarrow{\lambda}$	Rect.	67.82	4.86	75.28	0.02	0.01	0.05
	$\vec{\lambda}, \overleftarrow{\lambda}$	Polar	67.53	4.90	75.06	0.02	0.01	0.05
	$\vec{\nu}, \overleftarrow{\nu}$	Rect.	2.09	0.17	2.62	0.02	0.00	0.04
	$\vec{\nu}, \overleftarrow{\nu}$	Polar	1.57	0.18	2.11	0.02	0.00	0.04

All gaps are w.r.t. Clarabel$_{\text{quad}}$. †After dual feasibility restoration.

TABLE VI
GEOMETRIC MEANS OF COMPUTING TIME OF DCP AND SOLVERS

System	DCP	Mosek	Clarabel$_{\text{quad}}$
ieee14	4 ms	0.01 s	0.12 s
ieee118	21 ms	0.44 s	1.32 s
ieee300	15 ms	1.26 s	5.28 s
pegase1354	32 ms	10.19 s	33.14 s
pegase2869	66 ms	51.32 s	90.32 s

Mosek and Clarabel$_{\text{quad}}$ times are per instance, using one CPU core. DCP times are per batch of 512 instances, using a GPU.

Second, although Mosek's dual solutions may exhibit non-trivial violations, passing them through the dual completion steps yields overall high-quality dual bounds. The large maximum optimality gap of 0.18%, observed on the ieee300 system, is mostly likely caused by numerical difficulties, and emphasizes the importance of ensuring valid dual bounds. Furthermore, the original (possibly invalid) dual bound returned by Mosek, i.e., before repairing dual feasibility, is on average within 0.03% of the optimal value; this is compared to average gaps of at most 0.02% after feasibility recovery. *These findings highlight the broader value of the proposed dual completion procedure: it can be used in conjunction with an optimization solver, to obtain high-quality valid dual bounds.*

Third, the best DCP proxies provide dual solutions with valid dual bounds that achieve average gaps below 0.1% for the smaller systems (ieee14, ieee118), about 0.5% on the medium-size ieee300, and about 1.6% for the larger pegase1354 and pegase2869 systems. One striking observation on the larger Pegase systems is that DCP also achieves good worst-case optimality gaps, of about 2% gap. This is especially important, because this worst-case performance is only 25% higher than the average performance, in contrast to that of the smaller IEEE systems where the worst

gap may be 500x larger than the average gap.

Finally, Table VI reports mean computing times for DCP, Mosek and Clarabel$_{\text{quad}}$. The computing time of DCP is reported per batch of 512 instances on a GPU under the same conditions as previously described. The computing time for Mosek and Clarabel$_{\text{quad}}$ are per instance, using a single CPU core. Computing times for Mosek and Clarabel$_{\text{quad}}$ were recorded during the data-generation process, during which 24 instances were solved simultaneously on a single CPU node, which inevitably yields higher computing times because of limited thread availability (one core per instance), cache access, and potential CPU frequency slowdown because of high CPU usage. Nevertheless, the proxies significantly outperform Mosek in terms of computing times, especially for larger instances. For `pegase2869`, using the best DCP configuration, evaluating a batch of 512 instances takes an average 66ms on a GPU, whereas solving one instance takes 51.32s on a single CPU core. *Overall, the DCP proxies yield three orders of magnitude* speedups over Mosek. These speedups allow DCP to provide, for the first time, performance guarantees for AC-OPF proxies in real time.

VI. Conclusion

The paper has proposed, for the first time, dual optimization proxies that provide valid high-quality lower bounds for AC-OPF in milliseconds. The proposed dual conic proxies build on convex relaxations of AC-OPF and conic duality. A core contribution of the paper is the proposed dual feasibility completion that guarantees dual feasibility, thus providing valid dual bounds by leveraging fundamental properties of dual-optimal solutions. This dual feasibility completion is not restricted to ML settings: it can provide certified dual bounds given close-to-feasible solutions produced by an optimization solver. Extensive numerical experiments on power grids with up to 2869 buses have demonstrated that such proxies can produce high-quality bounds within very short time, improving computing times by three orders of magnitude compared to a state-of-the-art optimization solver. Relevant directions of future work include the design of enhanced completion methods to further improve proxy performance, and applying the DCP framework to DC-OPF and other convex relaxations of AC-OPF such as the SDP relaxation.

References

[1] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 630–637, Apr. 2020.

[2] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," *arXiv preprint arXiv:2104.12225*, 2021.

[3] S. Park and P. Van Hentenryck, "Self-supervised primal-dual learning for constrained optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4052–4060.

[4] W. Chen, M. Tanneau, and P. V. Hentenryck, "End-to-End Feasible Optimization Proxies for Large-Scale Economic Dispatch," *IEEE Transactions on Power Systems*, pp. 1–12, 2023.

[5] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2020, pp. 1–6.

[6] W. Huang, X. Pan, M. Chen, and S. H. Low, "DeepOPF-V: Solving AC-OPF Problems Efficiently," *IEEE Transactions on Power Systems*, vol. 37, no. 1, pp. 800–803, 2022.

[7] MISO, "Energy and operating reserve markets," 2022, Business Practices Manual Energy and Operating Reserve Markets.

[8] R. Nellikkath and S. Chatzivasileiadis, "Physics-Informed Neural Networks for Minimising Worst-Case Violations in DC Optimal Power Flow," in *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2021, pp. 419–424.

[9] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1–7.

[10] S. H. Low, "Convex relaxation of Optimal Power Flow—Part I: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014.

[11] ——, "Convex relaxation of Optimal Power Flow—Part II: Exactness," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 177–189, 2014.

[12] R. A. Jabr, "Radial distribution load flow using conic programming," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1458–1459, Aug 2006.

[13] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 6, pp. 383–392, 2008.

[14] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "The QC relaxation: A theoretical and computational study on optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3008–3018, 2015.

[15] D. K. Molzahn and I. A. Hiskens, "Moment-Based Relaxation of the Optimal Power Flow Problem," in *18th Power Syst. Comput. Conf. (PSCC)*, Aug. 18-22, 2014.

[16] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-hard," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 798–801, 2015.

[17] B. Kocuk, S. S. Dey, and X. A. Sun, "Strong SOCP relaxations for the Optimal Power Flow problem," *Operations Research*, vol. 64, no. 6, pp. 1177–1196, 2016.

[18] S. Gopinath, H. Hijazi, T. Weisser, H. Nagarajan, M. Yetkin, K. Sundar, and R. Bent, "Proving global optimality of ACOPF solutions," *Electric Power Systems Research*, vol. 189, p. 106688, 2020.

[19] F. Thams, A. Venzke, R. Eriksson, and S. Chatzivasileiadis, "Efficient Database Generation for Data-Driven Security Assessment of Power Systems," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 30–41, 2020.

[20] A. Venzke, D. K. Molzahn, and S. Chatzivasileiadis, "Efficient creation of datasets for data-driven power system applications," *Electric Power Systems Research*, vol. 190, p. 106614, 2021.

[21] F. Cengil, H. Nagarajan, R. Bent, S. Eksioglu, and B. Eksioglu, "Learning to accelerate globally optimal solutions to the AC Optimal Power Flow problem," *Electric Power Systems Research*, vol. 212, p. 108275, 2022.

[22] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*. IEEE, 2019, pp. 1–6.

[23] X. Pan, T. Zhao, M. Chen, and S. Zhang, "Deep-OPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1725–1735, 2020.

[24] W. Chen, S. Park, M. Tanneau, and P. Van Hentenryck, "Learning optimization proxies for large-scale security-constrained economic dispatch," *Electric Power Systems Research*, vol. 213, p. 108566, 2022.

[25] M. Kim and H. Kim, "Projection-aware Deep Neural Network for DC Optimal Power Flow Without Constraint Violations," in *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2022, pp. 116–121.

[26] M. Li, S. Kolouri, and J. Mohammadi, "Learning to solve optimization problems with hard linear constraints," *IEEE Access*, 2023.

[27] M. Chatzos, T. W. Mak, and P. Van Hentenryck, "Spatial network decomposition for fast and scalable AC-OPF learning," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 2601–2612, 2021.

[28] X. Pan, M. Chen, T. Zhao, and S. H. Low, "DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems," *IEEE Systems Journal*, vol. 17, no. 1, pp. 673–683, 2022.

[29] T. W. Mak, M. Chatzos, M. Tanneau, and P. V. Hentenryck, "Learning Regionally Decentralized AC Optimal Power Flows with ADMM," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023.

[30] S. Park, W. Chen, T. W. Mak, and P. Van Hentenryck, "Compact optimization learning for AC optimal power flow," *arXiv preprint arXiv:2301.08840*, 2023.

[31] R. Nellikkath and S. Chatzivasileiadis, "Physics-Informed Neural Networks for AC Optimal Power Flow," *Electric Power Systems Research*, vol. 212, p. 108412, 2022.

[32] W. Huang and M. Chen, "DeepOPF-NGT: Fast No Ground Truth Deep Learning-Based Approach for AC-OPF Problems," in *ICML 2021 Workshop Tackling Climate Change with Machine Learning*, 2021.

[33] D. Owerko, F. Gama, and A. Ribeiro, "Unsupervised optimal power flow using graph neural networks," *arXiv preprint arXiv:2210.09277*, 2022.

[34] S. Chevalier and S. Chatzivasileiadis, "Global performance guarantees for neural network models of ac power flow," 2023.

[35] S. Chevalier, I. Murzakhanov, and S. Chatzivasileiadis, "GPU-Accelerated Verification of Machine Learning Models for Power Systems," 2023.

[36] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[37] S. Babaeinejadsarookolaee *et al.*, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," *arXiv:1908.02788*, 2019.

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

[39] M. ApS, *MOSEK Optimizer API for Julia 10.1.12*, 2022. [Online]. Available: https://docs.mosek.com/10.1/juliaapi/index.html

[40] P. Goulart and Y. Chen, *Clarabel.jl*, 2024. [Online]. Available: https://oxfordcontrol.github.io/ClarabelDocs/stable/

[41] D. Bienstock and M. Villagra, "Accurate and Warm-Startable Linear Cutting-Plane Relaxations for ACOPF," 2024. [Online]. Available: https://optimization-online.org/2024/02/accurate-and-warm-startable-linear-cutting-plane-relaxations-for-acopf/

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

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] PACE, *Partnership for an Advanced Computing Environment (PACE)*, 2017. [Online]. Available: http://www.pace.gatech.edu

[45] A. Oustry, C. D'Ambrosio, L. Liberti, and M. Ruiz, "Certified and accurate SDP bounds for the ACOPF problem," *Electric Power Systems Research*, vol. 212, p. 108278, 2022.