Title

Presolving convexified optimal power flow with mixtures of gradient experts

Journal

Energy and AI, 22

ISSN

2666-5468

Authors

Bose, Shourya

Chen, Kejun

Zhang, Yu

Publication Date

2025-12-01

DOI

10.1016/j.egyai.2025.100609

Presolving Convexified Optimal Power Flow with Mixtures of Gradient Experts

Shourya Bose [a], Kejun Chen [a], Yu Zhang [a]

[a] *University of California Santa Cruz, 1156 High St., Santa Cruz, CA 95064, USA*

Abstract

Convex relaxations and approximations of the optimal power flow (OPF) problem have gained significant research and industrial interest for planning and operations in electric power networks. One approach for reducing their solve times is *presolving* which eliminates constraints from the problem definition, thereby reducing the burden of the underlying optimization algorithm. To this end, we propose a presolving framework for convexified optimal power flow (C-OPF) problems, which uses a novel deep learning-based architecture called `MoGE` (Mixture of Gradient Experts). In this framework, problem size is reduced by learning the mapping between C-OPF parameters and optimal dual variables (the latter being representable as gradients), which is then used to screen constraints that are non-binding at optimum. The validity of using this presolve framework across arbitrary families of C-OPF problems is theoretically demonstrated. We characterize generalization in `MoGE` and develop a post-solve recovery procedure to mitigate possible constraint classification errors. Using two different C-OPF models, we show via simulations that our framework reduces solve times by upto 34% across multiple PGLIB and MATPOWER test cases, while providing an identical solution as the full problem.

Keywords: optimal power flow, convex relaxation, convex optimization, constraint screening, input convex neural network, monotone gradient network, physics-informed neural network, constraint qualification, data augmentation, deep learning, artifical intelligence, energy optimization

1. Introduction

The Optimal Power Flow (OPF) determines the optimal settings for control variables in an electric power network. Its primary goal is to minimize the total generation cost while satisfying various operational constraints (e.g., generation

⋆This work was partially supported by the 2023 CITRIS Interdisciplinary Innovation Program (I2P).

Email addresses: `shbose@ucsc.edu` (Shourya Bose), `kchen158@ucsc.edu` (Kejun Chen), `zhangy@ucsc.edu` (Yu Zhang)

limits, voltage limits, and thermal limits). OPF is solved by different entities such as independent system operators [1], energy markets [2], and microgrid operators [3] as a part of their planning and operational procedures. Increasing stochastic factors such as load demand fluctuations and varying generation from renewable distributed energy resources (DERs) necessitate frequent re-solving of OPF, which in turn calls for time-efficient solution methods.

Convexified formulations of AC optimal power flow (ACOPF), which we denote by C-OPFs, have recently gained significant research and industrial attention. C-OPF formulations benefit from theoretical guarantees of global optimality and the availability of mature commercial solvers. Popular C-OPF formulations include DC optimal power flow (DCOPF) [4], convexified DistFlow OPF [5, 6, 7], semidefinite relaxation (SDR) [8, 9, 10], second-order cone programming (SOCP) relaxation [11], quadratic-convex relaxation [12], McCormick relaxation [13], etc. A comprehensive review of convex relaxations and approximations of OPF can be found in [14].

1.1. Motivation and Scope

In this paper, we aim to reduce the solution time of C-OPF solvers using methods that are agnostic to the underlying optimization algorithm. This is achieved by using the presolving approach of *Constraint Screening (CS)*, which identifies and eliminates constraints that are non-binding at optimum, thereby reducing problem size and accelerating solution times. This is done by learning the mapping between problem parameters and optimal dual variables using our proposed `MoGE` architecture, which allows identifying binding constraints using the Kahrush-Kuhn-Tucker (KKT) conditions. These problem parameters include load demands and possibly other quantities which vary between different C-OPF instances.

We restrict our attention to C-OPFs rather than the nonconvex ACOPF due to several reasons. C-OPFs are widely used as a faithful approximation of ACOPF [15], with the workhorse DCOPF being a linear, and therefore convex approximation of ACOPF. Moreover, SOCP- and SDP-based C-OPFs allow recovery of optimal ACOPF solutions under certain circumstances [16, 17]. Lastly, convex relaxations of ACOPF provide a lower bound to the optimal ACOPF objective value, which is useful in branch-and-bound algorithms for solving mixed-integer formulations of ACOPF [18].

We also highlight that CS differs fundamentally from many other methods to accelerate OPF. For example, methods based on machine learning [19, 20] learn the mapping from input parameters to a partial set of OPF variables, with the remaining variables computed algorithmically. Similarly, our previous works [21, 22] use physics-informed neural networks and gradient learning to directly learn OPF solutions, or parts thereof. Some other methods for accelerating OPF include graph neural networks [23], evolutionary algorithms [24, 25], simulated annealing [26], and particle swarm optimization [27]. The aforementioned methods, while benefiting from fast solutions, lack provable guarantees of local optimality or even feasibility of the resulting solutions due to generalization errors inherent to machine learning techniques. On the other hand, the

present work identifies CS as an approach which can result in provably feasible and optimal solutions, even with the use of machine learning.

1.2. Prior Work

Earliest works on the removal of redundant and non-binding constraints are for linear programs [28, 29]. Recent domain-specific CS methods for power systems are based on geometry of the feasible set or machine learning techniques. CS methods based on feasible set geometry analyze the constraints *a priori* to generate a list of constraints which are provably non-binding at the optimum. In this category, various algorithms have been proposed to eliminate redundant line flow constraints for DCOPF formulations, based on solving auxiliary optimization problems (see [30, 31, 32]). Similarly, different algorithms have been developed to accelerate the branch-and-bound algorithm for unit commitment by eliminating network constraints (see [33, 34]). However, geometry-based methods require expensive re-computations for every *test case* (for example a 30-bus system vs. a 141-bus system), which involves solving auxiliary optimization problems whose count is proportional to the number of constraints being targeted for elimination. Furthermore, since these methods aim to remove non-binding constraints over *all* possible instances, the actual number of constraints removed are fewer.

Machine learning-based CS methods leverage historical or synthetic OPF datasets to train learning models, with load demands typically included as key problem parameters. For example, Deka and Misra [35] and Hasan and Kargarian [36] train deep neural network (DNN) based classifiers to learn the mapping between problem parameters and the binding constraints directly. These works fall under the umbrella category of *active-set methods*, which aim to learn the optimal set of binding constraints from 2^M possible configurations of a problem that contains M candidate constraints. Dual variables play a pivotal role in DNN applications for OPF, as their optimal values reveal the binding status of corresponding constraints [37, 38]. Chen, Zhang, Chen, and Zhang [39] learn optimal dual variables with DNNs to solve DCOPF via solving the dual problem. Nellikkath and Chatzivasileiadis [40] leverage the KKT system to improve the robustness of deep learning for learning optimal solutions.

1.3. Contribution

Based on learning the dual variables, we propose a CS method to reduce the size of C-OPF problems in this paper. We first present a generic representation which can denote any arbitrary C-OPF as a parametric convex optimization problem, with parameters representing values such as load demands, thermal limits, etc. We develop a novel neural network architecture called ***`MoGE`*** *(Mixture of Gradient Experts)* to learn the mapping from parameters to optimal dual variables. Once trained, `MoGE` predicts optimal dual variables for new C-OPF instances, enabling the identification of non-binding constraints via complementary slackness. Removing these non-binding constraints reduces the problem size, thereby accelerating the solution process. In addition, we address the issue

of potentially misclassified constraints by incorporating a recovery process. Any identified violated constraints after solving the reduced C-OPF are retroactively added to the active constraints, and the problem is re-solved. We harness the convexity of C-OPF to guarantee that even with mispredicted constraints, a finite number of re-solves are required to achieve an exact solution.

Our proposed approach presents significant advantages over existing active-set methods such as [35, 36]. We use dual variables as a proxy to identify binding constraints, which makes our method *physics-informed*. We also address weaknesses inherent in other dual variable-based methods such as [39, 40]. First, our framework can handle nonlinear convex constraints, necessitating an analysis of the existence and uniqueness of the dual solution, as well as the validity of using dual variables for constraint classification. We provide the necessary analyses to support this. Second, previous methods lack provable robustness against mispredictions of dual variables, instead only offering theoretical generalization guarantees which are difficult to quantify for practical C-OPF cases. We address this issue through our recovery procedure. Finally, we address practical issues with implementation such as dataset augmentation for efficient utilization of data, and perform extensive simulations on a variety of test cases from the PGLIB OPF [41] and MATPOWER [42] libraries to validate the merits of our proposed method.

1.4. Organization and Notation

The paper is structured as follows. Section 2 introduces a standard form of C-OPF that encompasses various OPF models, with the quadratic-convex (QC) and convexified DistFlow (CDF) relaxations of ACOPF serving as specific examples. Section 3 begins with an exploration of the analytical properties of C-OPF. Subsequently, we establish strong duality results for QC-OPF and CDF-OPF, which guarantees uniqueness of dual solutions. This analysis is then generalized into a test which can be performed for arbitrary families of C-OPF. This section also details the design and training of `MoGE`, along with the data generation process and the importance of dense sampling for the same. Section 4 conducts simulations to comparatively analyze the proposed method against other CS approaches. The conclusion is presented in Section 5, accompanied by key proofs provided in the Appendix.

Notation: Matrices and vectors are depicted in bold typeface. The n-dimensional real space is denoted by $\mathbb{R}^n$. $[n]$ denotes the set $\{1, \ldots, n\}$. For a vector $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a}_i$ is its i^{th} element while $[\mathbf{a}]_{\mathcal{S}}$ denotes the subvector corresponding to a set $\mathcal{S} \subseteq [n]$. $|\mathcal{S}|$ is the cardinality of set $\mathcal{S}$. $\mathbf{a} \preceq \mathbf{b}$ denotes elementwise inequality between the two vectors, while $(\mathbf{a}, \mathbf{b})$ denotes their concatenation. $\nabla f(\mathbf{x})$ denotes the gradient or Jacobian matrix of the function $f(\mathbf{x})$ while $\nabla^2 f(\mathbf{x})$ is its Hessian. $\mathbf{e}_i$ is the i^{th} canonical vector and $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\text{conv}\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \triangleq \{\sum_{i=1}^n \boldsymbol{\alpha}_i \mathbf{x}_i \mid \boldsymbol{\alpha} \succeq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\alpha} = 1\}$ denotes the convex hull of a finite set $\{\mathbf{x}_i\}_{i=1}^n$. $\mathtt{j} = \sqrt{-1}$ is the imaginary unit.

2. C-OPF as Parametric Convex Optimization

In this section, we represent C-OPF as a parametric convex optimization problem. This allows us to generate a well-defined mapping between problem parameters and optimal dual variables, which can then be used for CS. In general, constraints for any OPF problem can broadly be categorized into two types: the power flow equations and other operational constraints. The power flow model remains fixed across problem instances, while operational constraints, such as nodal power balance and thermal limits, may vary across problem instances. These constraints' bounds serve as problem parameters in our formulation.

2.1. Generic C-OPF Formulation

Concretely, we represent any C-OPF as the following parametric optimization problem:

$$
\begin{aligned}
\mathcal{V}(\boldsymbol{\gamma},\boldsymbol{\xi}) \triangleq \min_{\mathbf{x}} \quad & f(\mathbf{x}) && && \text{(1a)}\\
\text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \preceq \mathbf{0} \;\; (\boldsymbol{\lambda}), \quad && \mathbf{h}(\mathbf{x}) = \mathbf{0} \;\; (\boldsymbol{\mu}) && \text{(1b)}\\
& \tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma} \;\; (\tilde{\boldsymbol{\lambda}}), \quad && \tilde{\mathbf{h}}(\mathbf{x}) = \boldsymbol{\xi} \;\; (\tilde{\boldsymbol{\mu}}) && \text{(1c)}\\
& \underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}} \;\; (\underline{\boldsymbol{\nu}}, \bar{\boldsymbol{\nu}}). && && \text{(1d)}
\end{aligned}
$$

Here, $\mathbf{x} \in \mathbb{R}^n$ is the decision variable whose exact description depends on the underlying power flow model. The convex objective function $f : \mathbb{R}^n \mapsto \mathbb{R}$ typically represents the total generation cost or power line losses over the network. Constraint functions $\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^L$ and $\mathbf{h} : \mathbb{R}^n \mapsto \mathbb{R}^M$ denote the power flow model, while $\tilde{\mathbf{g}} : \mathbb{R}^n \mapsto \mathbb{R}^{\tilde{L}}$ and $\tilde{\mathbf{h}} : \mathbb{R}^n \mapsto \mathbb{R}^{\tilde{M}}$ denote all other operational constraints. The dual variables for each block of constraints are represented alongside in parentheses. We use the terminology that $\mathcal{V}$ forms a *family* of C-OPF problems under a given power flow model, while its specific values for a given topology and different $(\boldsymbol{\gamma},\boldsymbol{\xi})$ form *instances*.

The bounds for the operational constraints are captured by the right-hand side parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$, which are used to parameterize the problem. For the parameter space, we consider open, nonempty subsets $\Gamma \subset \mathbb{R}^{\tilde{L}}$ and $\Xi \subset \mathbb{R}^{\tilde{M}}$ such that problem (1) is feasible for all $(\boldsymbol{\gamma},\boldsymbol{\xi}) \in \Gamma \times \Xi$.

Definition 1. *The feasible power flow set is defined as*

$$\mathcal{X}_{\text{pf}} \triangleq \{\mathbf{x} : \mathbf{g}(\mathbf{x}) \preceq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\}.$$

Definition 2. *The active set of the inequality constraint* $\mathbf{g}(\mathbf{x}) \preceq \mathbf{0}$ *at point* $\mathbf{x}$ *is defined as*

$$\mathcal{A}_{\mathbf{g}}(\mathbf{x}) \triangleq \{i \in [L] : \mathbf{g}_i(\mathbf{x}) = 0\}.$$

Table 1: Representing QC-OPF (2) as parametric convex optimization problem (1), where the box constraint (1d) corresponds to constraints (2g), (2h), and (2i).

Variables	Constraints	Parameters
$\mathbf{x}$: $\{P_k^g, Q_k^g\}_{k\in\mathcal{G}}$, $\{P_{ij}, Q_{ij}\}_{(i,j)\in\mathcal{E}\cup\mathcal{E}^R}$, $\{W_{ij}^{\mathrm{R}}, W_{ij}^{\mathrm{I}}\}_{(i,j)\in\mathcal{E}}, \{W_{ii}\}_{i\in\mathcal{N}}$	$g(\mathbf{x}), \tilde{g}(\mathbf{x})$: (2c)–(2d), (2f) $h(\mathbf{x}), \tilde{h}(\mathbf{x})$: (2b), (2e)	$\boldsymbol{\gamma}$: $\{\bar{S}_{ij}^2\}_{(i,j)\in\mathcal{E}}, \{\bar{S}_{ij}^2\}_{(i,j)\in\mathcal{E}}$ $\boldsymbol{\xi}$: $\{P_i^d, Q_i^d\}_{i\in\mathcal{N}}$

In the sequel, we show how the quadratic-convex relaxation to OPF (QC-OPF) [12] can be represented in the form of (1). QC relaxation is generally tighter than SOCP relaxation and faster to compute than SDP relaxation [11, 12]. While QC-OPF serves as our demonstrative example, the proposed framework and analysis are applicable to other C-OPF models. For example, the present framework can also be applied to the convexified DistFlow OPF model (CDF-OPF), which is a nonlinear representation of power flows in low-voltage radial power networks. We relegate said analysis to Appendix A.

2.2. QC-OPF

QC-OPF is a convex relaxation to the ACOPF for meshed networks, possibly multiple generators on each bus, and transformers and phase shifters on each line. Let $(\mathcal{N}, \mathcal{E})$ denote the directed graph representing the power network, and $\mathcal{E}^R$ the edges with the reversed orientation. Let $N = |\mathcal{N}|$ and $E = |\mathcal{E}|$ denote the total number of buses and power lines. Each bus $i \in \mathcal{N}$ is associated with the indices $\mathcal{G}_i$ of generators attached to it, real and reactive demands P_i^d, Q_i^d, line charging admittance $g_i^s - \mathrm{j}b_i^s$, and minimum (maximum) voltage magnitude $\underline{V}_i$ ($\bar{V}_i$). In addition, the real and reactive power outputs of the k^{th} generator are denoted by P_k^g and Q_k^g, respectively. Set $\mathcal{G} \triangleq \cup_{i\in\mathcal{N}}\mathcal{G}_i$ collects all generation units. For each branch $(i,j) \in \mathcal{E} \cup \mathcal{E}^R$ we have the thermal limit $\bar{S}_{ij}$, and the minimum and maximum phase angle differences (PAD) such that $-\frac{\pi}{2} < \underline{\theta}_{ij} < \bar{\theta}_{ij} < \frac{\pi}{2}$. Each branch is represented by a Π-model with the corresponding line matrix

$$\begin{bmatrix} y_{ij}^{\mathrm{ff}} & y_{ij}^{\mathrm{ft}} \\ y_{ij}^{\mathrm{tf}} & y_{ij}^{\mathrm{tt}} \end{bmatrix} \triangleq \begin{bmatrix} \left(y_{ij} + \mathrm{j}\frac{b_{ij}^c}{2}\right)\frac{1}{|T_{ij}|^2} & -\frac{y_{ij}}{T_{ij}^*} \\ -\frac{y_{ij}}{T_{ij}} & y_{ij} + \mathrm{j}\frac{b_{ij}^c}{2} \end{bmatrix},$$

where we have line admittance y_{ij}, total charging susceptance b_{ij}^c, and transformer parameter $T_{ij} = \tau_{ij}\angle\theta_{ij}^{\mathrm{shift}}$.

Let V_i denote the voltage phasor at bus i and define $W_{ij} = W_{ij}^{\mathrm{R}} + \mathrm{j}W_{ij}^{\mathrm{I}} \triangleq V_iV_j^*$. To this end, the QC-OPF formulation is given as follows.

$$\min \sum_{k\in\mathcal{G}} c_{k,2}\left(P_k^g\right)^2 + c_{k,1}P_k^g + c_{k,0} \tag{2a}$$

subject to:

$$\left.\begin{array}{l} P_{ij}-\mathrm{Re}\left(y_{ij}^{\mathrm{ff}}\right) W_{ii}-\mathrm{Re}\left(y_{ij}^{\mathrm{ft}}\right) W_{ij}^{\mathrm{R}}-\mathrm{Im}\left(y_{ij}^{\mathrm{ft}}\right) W_{ij}^{\mathrm{I}}=0 \\ Q_{ij}+\mathrm{Im}\left(y_{ij}^{\mathrm{ff}}\right) W_{ii}-\mathrm{Re}\left(y_{ij}^{\mathrm{ft}}\right) W_{ij}^{\mathrm{I}}+\mathrm{Im}\left(y_{ij}^{\mathrm{ft}}\right) W_{ij}^{\mathrm{R}}=0 \\ P_{ji}-\mathrm{Re}\left(y_{ij}^{\mathrm{tt}}\right) W_{jj}-\mathrm{Re}\left(y_{ij}^{\mathrm{tf}}\right) W_{ij}^{\mathrm{R}}+\mathrm{Im}\left(y_{ij}^{\mathrm{tf}}\right) W_{ij}^{\mathrm{I}}=0 \\ Q_{ji}+\mathrm{Im}\left(y_{ij}^{\mathrm{tt}}\right) W_{jj}+\mathrm{Re}\left(y_{ij}^{\mathrm{tf}}\right) W_{ij}^{\mathrm{I}}+\mathrm{Im}\left(y_{ij}^{\mathrm{tf}}\right) W_{ij}^{\mathrm{R}}=0 \end{array}\right\} \forall (i,j)\in\mathcal{E} \tag{2b}$$

$$(W_{ij}^{\mathrm{R}})^2+(W_{ij}^{\mathrm{I}})^2-W_{ii}W_{jj}\leq 0,\ \forall (i,j)\in\mathcal{E} \tag{2c}$$

$$\tan\left(\underline{\theta}_{ij}\right) W_{ij}^{\mathrm{R}}\leq W_{ij}^{\mathrm{I}}\leq \tan\left(\bar{\theta}_{ij}\right) W_{ij}^{\mathrm{R}},\ \forall (i,j)\in\mathcal{E} \tag{2d}$$

$$\left.\begin{array}{l} \left(\sum\limits_{k\in\mathcal{G}_i} P_k^g\right)-g_i^s W_{ii}-\sum\limits_{(i,j)\in\mathcal{E}\cup\mathcal{E}^R} P_{ij}=P_i^d \\ \left(\sum\limits_{k\in\mathcal{G}_i} Q_k^g\right)+b_i^s W_{ii}-\sum\limits_{(i,j)\in\mathcal{E}\cup\mathcal{E}^R} Q_{ij}=Q_i^d \end{array}\right\},\ \forall i\in\mathcal{N} \tag{2e}$$

$$P_{ij}^2+Q_{ij}^2\leq \bar{S}_{ij}^2,\quad P_{ji}^2+Q_{ji}^2\leq \bar{S}_{ij}^2,\ \forall (i,j)\in\mathcal{E} \tag{2f}$$

$$\underline{P}_k^g\leq P_k^g\leq \bar{P}_k^g,\quad \underline{Q}_k^g\leq Q_k^g\leq \bar{Q}_k^g,\ \forall k\in\mathcal{G} \tag{2g}$$

$$\underline{V}_i^2\leq W_{ii}\leq \bar{V}_i^2,\ \forall i\in\mathcal{N} \tag{2h}$$

$$|W_{ij}^{\mathrm{R}}|\leq \bar{V}_i\bar{V}_j,\quad |W_{ij}^{\mathrm{I}}|\leq \bar{V}_i\bar{V}_j,\ \forall (i,j)\in\mathcal{E}. \tag{2i}$$

In the above, the objective (2a) is a convex quadratic function of real power generation cost. (2b) describes the forward and reverse flows on each line, while (2c) denotes the SOCP relaxation of the cross-term dependence. (2d) enforces the PAD limits on each branch. (2e) represents the nodal power balance, while line flow constraints (2f) enforce the thermal limits. Finally, (2g), (2h), and (2i) represent the upper and lower bounds for the decision variables.

We can represent the QC-OPF (2) in the generic formulation (1), as shown in Table 1. The parameterized inequality $\tilde{\mathbf{g}}(\mathbf{x})\preceq\boldsymbol{\gamma}$ contains the line flow constraints and the equality $\tilde{\mathbf{h}}(\mathbf{x})=\boldsymbol{\xi}$ contains the power balance equations. The former by is parameterized by the thermal limits, and the latter by real and reactive load demands.

3. Constraint Screening for C-OPF

We divide this section into two parts. The first part provides analysis for generalized C-OPF of the form (1). In the second part, we present the proposed accelerated solution based on constraint screening.

3.1. Analytical Properties of C-OPF

We start with a standard result of the value function $\mathcal{V}(\boldsymbol{\gamma},\boldsymbol{\xi})$.

Lemma 1 (Convexity of value function [43, pp. 250])**.** *$\mathcal{V}(\boldsymbol{\gamma},\boldsymbol{\xi})$ is jointly convex in $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ over any convex subset of $\Gamma\times\Xi$.*

Corollary 1 (Feasibility in the parameter space)**.** *Assume problem (1) is feasible for each point in a finite set $\{(\boldsymbol{\gamma}^{(k)},\boldsymbol{\xi}^{(k)})\}_{k\in[K]}$. Then, the problem remains feasible for any point $(\boldsymbol{\gamma},\boldsymbol{\xi})\in\mathrm{conv}\{(\boldsymbol{\gamma}^{(1)},\boldsymbol{\xi}^{(1)}),\ldots,(\boldsymbol{\gamma}^{(K)},\boldsymbol{\xi}^{(K)})\}$.*

This corollary supports our data generation Algorithm 1, and has a simple proof. $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$ is finite because the problem is feasible for each point $(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\xi}^{(k)})$. Due to the convexity of $\mathcal{V}$, we have $\mathcal{V}(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\xi}}) \leq \sum_k \boldsymbol{\alpha}_k \mathcal{V}(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\xi}^{(k)}) < \infty$, where $(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\xi}}) \triangleq \sum_k \boldsymbol{\alpha}_k (\boldsymbol{\gamma}^{(k)}, \boldsymbol{\xi}^{(k)})$ for any $\boldsymbol{\alpha} \succeq \mathbf{0}$ and $\mathbf{1}^\top \boldsymbol{\alpha} = 1$. This confirms that the problem is feasible for $(\tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\xi}})$. The partial Lagrangian function of (1) can be written as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}, \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) \triangleq f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^\top \mathbf{h}(\mathbf{x}) + \tilde{\boldsymbol{\lambda}}^\top (\tilde{\mathbf{g}}(\mathbf{x}) - \boldsymbol{\gamma}) + \tilde{\boldsymbol{\mu}}^\top (\tilde{\mathbf{h}}(\mathbf{x}) - \boldsymbol{\xi}).$$

The dual problem is

$$\sup_{\boldsymbol{\lambda} \succeq \mathbf{0}, \tilde{\boldsymbol{\lambda}} \succeq \mathbf{0}, \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}} D(\boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}, \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}), \tag{3}$$

with the dual objective

$$\begin{aligned} D(\boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}, \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) &\triangleq \inf_{\underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \tilde{\boldsymbol{\lambda}}, \boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) \\ &= -\tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\gamma} - \tilde{\boldsymbol{\mu}}^\top \boldsymbol{\xi} + \inf_{\underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}}} \big\{ f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{x}) \\ &\qquad + \boldsymbol{\mu}^\top \mathbf{h}(\mathbf{x}) + \tilde{\boldsymbol{\lambda}}^\top \tilde{\mathbf{g}}(\mathbf{x}) + \tilde{\boldsymbol{\mu}}^\top \tilde{\mathbf{h}}(\mathbf{x}) \big\}. \end{aligned} \tag{4}$$

If strong duality holds (meaning the optimal values of the primal and dual problems are equal), the structure in (4) indicates that gradients of $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$ exist and can be expressed as functions of the optimal dual solutions. Strong duality is not guaranteed in general, though it always holds for feasible linear programs [44, Chap. 6]. For convex problems, various results establish conditions under which strong duality holds. These conditions are called constraint qualifications (a.k.a. regularity conditions). Constraint qualifications guarantee that a constrained minimizer satisfies the well-defined KKT system.

Next, we show that the special structure of QC-OPF (2) ensures strong duality by satisfying the *linear independence constraint qualification (LICQ)*. We then generalize this finding to a test which can be used on arbitrary C-OPF models.

Lemma 2 (Fixed LICQ)**.** *Given QC-OPF* (2)*, for all* $\tilde{\mathbf{x}} \in \mathcal{X}_{\text{pf}}$ *the LICQ holds with respect to the "fixed" constraints* $\mathbf{g}$ *and* $\mathbf{h}$*, i.e., the Jacobian matrix of the binding constraints is full row rank*[1]*:*

$$\operatorname{rank}\left(\mathbf{J}(\tilde{\mathbf{x}}) \triangleq \begin{bmatrix} \nabla [\mathbf{g}(\tilde{\mathbf{x}})]_{\mathcal{A}_{\mathbf{g}}(\tilde{\mathbf{x}})} \\ \nabla \mathbf{h}(\tilde{\mathbf{x}}) \end{bmatrix} \right) = M + |\mathcal{A}_{\mathbf{g}}(\tilde{\mathbf{x}})|.$$

The proof is provided in Appendix B. This lemma allows us to extend LICQ to all constraints of QC-OPF over the entire parameter space. This is established in the following theorem.

[1]For simplicity, full row rank is referred to as full rank hereinafter.

Theorem 1 (Genericity of LICQ)**.** *For every feasible point of QC-OPF* (2) *and for almost everywhere in the parameter space* $\Gamma \times \Xi$*, the LICQ holds with respect to all constraints* $\mathbf{g}, \mathbf{h}, \tilde{\mathbf{g}}, \tilde{\mathbf{h}}$ *upon satisfaction of Lemma 2. Thus, strong duality holds for QC-OPF.*

The proof is provided in Appendix C. For the convex QC-OPF, strong duality implies $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) = -\tilde{\boldsymbol{\lambda}}^{*\top} \boldsymbol{\gamma} - \tilde{\boldsymbol{\mu}}^{*\top} \boldsymbol{\xi} + W$, where W denotes additional terms not involving $\boldsymbol{\gamma}$ or $\boldsymbol{\xi}$ (cf. (4)). Lemma 2 and Theorem 1 can be converted into a generalized test for strong duality for arbitrary C-OPF models beyond QC-OPF. It is sketched in the following remark and is demonstrated for CDF-OPF in Appendix A.

Remark 1 (Strong Duality Test for Arbitrary C-OPF)**.** *For any C-OPF which can be written in the form* (1)*, if the individual components of the functions* $\mathbf{g}$*,* $\mathbf{h}$*,* $\tilde{\mathbf{g}}$*, and* $\tilde{\mathbf{h}}$ *are smooth (i.e. they belong to class* C^∞*) and the fixed constraints* $\mathbf{g}$ *and* $\mathbf{h}$ *satisfy Lemma 2, then strong duality holds almost everywhere on the open set* $\Gamma \times \Xi$ *representing the parameter space. Since this test only requires checking a rank condition for fixed constraints, it can be used to certify an C-OPF family at once.*

LICQ (correspondingly, strong duality) ensures the existence and uniqueness of Lagrange multipliers satisfying the KKT system [45, Thm. 2]. The closed form of $\nabla_{\boldsymbol{\gamma}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$ and $\nabla_{\boldsymbol{\xi}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$ follows immediately, as given below.

Proposition 1 (Gradients of value function)**.** *Given QC-OPF* (2)*, for almost every* $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \in \Gamma \times \Xi$ *we have*

$$\nabla_{\boldsymbol{\gamma}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) = -\tilde{\boldsymbol{\lambda}}^*, \quad \nabla_{\boldsymbol{\xi}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) = -\tilde{\boldsymbol{\mu}}^*, \tag{5}$$

where $\tilde{\boldsymbol{\lambda}}^*$ *and* $\tilde{\boldsymbol{\mu}}^*$ *are the unique optimal solution to problem* (3)*.*

An equivalent result holds for CDF-OPF as shown in Appendix A, as well as any other C-OPF family which satisfy the criteria outlined in Remark 1. Since the optimal dual solution serves as the gradients of $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$, we can develop a customized neural network to learn these gradients and predict binding constraints for CS, as demonstrated in subsection 3.3. Note that the dual solution-based DCOPF solvers (e.g., [39]) can be considered as a special case of our framework.

3.2. Strict Complementarity

In this part, we justify using dual variables to remove non-binding constraints at the optimum by assuming the following condition for C-OPF.

Assumption 1 (Strict complementarity)**.** *Given C-OPF* (1)*, the strict complementarity condition holds for* $\tilde{\mathbf{g}}$*, i.e., for any optimal primal-dual solution we have*

$$\tilde{\boldsymbol{\lambda}}_i^* > 0, \text{ for all } i \in \mathcal{A}_{\tilde{\mathbf{g}}}(\mathbf{x}^*). \tag{6}$$

The strict complementarity condition asserts that all binding inequality constraints have strictly positive multipliers (i.e., no degenerate inequalities). This condition has been assumed in previous works on OPF [38, Assumption 1], and is observed empirically in our simulations. However, degeneracy has been observed to occur in certain C-OPF models used in energy markets [46]. In case the C-OPF is a conic convex problem, degeneracy can be removed by slightly perturbing the parameter and resolving. This stems from the fact that strict complementarity is a *generic* property of conic convex problems [47, Theorem 5]. We relegate a full analysis of CS with degenerate solutions to future works.

Proposition 2 (Constraint reduction equivalence)**.** *Suppose it is known* a priori *that* $\tilde{\boldsymbol{\lambda}}_i^* = 0$ *for* $i \in \mathcal{I} \subseteq [\tilde{L}]$*. Then, replacing constraint* $\tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}$ *in* (1) *with* $[\tilde{\mathbf{g}}(\mathbf{x})]_{[\tilde{L}]\setminus\mathcal{I}} \preceq \boldsymbol{\gamma}_{[\tilde{L}]\setminus\mathcal{I}}$ *results in an equivalent problem.*

The proof is straightforward. Any optimal solution $\mathbf{x}^*$ of the original problem satisfies the KKT system for the reduced problem, and vice versa. Consequently, the optimal solutions to both convex problems are identical. It is worth noting that Proposition 2 does not hold for nonconvex problems in general. Consider the problem

$$\min_{x \leq 0} \quad (4x^2 + x + 1)(x^2 - 1),$$

where LICQ holds for all points. The KKT system is

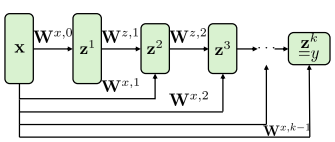
$$\begin{gathered} \nabla_x \left((4x^2 + x + 1)(x^2 - 1) + \lambda x \right) = 0 \\ x \leq 0, \quad \lambda \geq 0, \quad \lambda x = 0. \end{gathered}$$

We can verify that $x^* = -0.6267, \lambda^* = 0$ is the globally optimal solution. Excluding the constraint $x \leq 0$ given $\lambda^* = 0$, the optimal solution to the unconstrained problem becomes $x^* = 0.6043$. Therefore, the reduced problem is not equivalent to the original one.

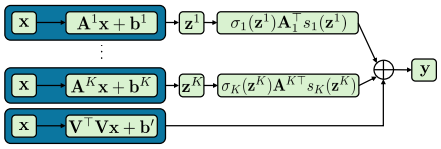
3.3. Constraint Screening via MoGE

In this section, we present the acceleration method for C-OPF by removing non-binding constraints from $\tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}$. Our strategy involves training a model which learns the gradients of $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$, i.e. the mapping $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \mapsto (-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*)$ in supervised learning fashion. Since this mapping is the gradient of the value function (Proposition 1), it motivates the "gradient" terminology used throught this paper. Any new problem instance is sped up by removing non-binding constraints identified from the predicted value of $\tilde{\boldsymbol{\lambda}}^*$ using Assumption 1.

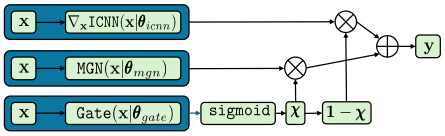
We aim to use machine learning models suitable for approximating gradients of convex functions, and consider two possible approaches. The first approach, used by input convex neural networks (ICNNs) [48], approximates the scalar-valued mapping $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \mapsto \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})$. Gradients are then generated by calculating $\nabla\mathcal{V}$ using automatic differentiation (AD) during both training and inference. The second approach, utilized by monotone gradient networks (MGNs) [49], directly learns the mapping $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \mapsto (-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*)$ without using AD.



(a) $\mathbf{y} = \texttt{ICNN}(\mathbf{x}|\boldsymbol{\theta}_{icnn})$, with nonnegative weights $\{\mathbf{W}^{z,i}\}_{i=1}^{k-1}$ ensuring convexity of $\mathbf{y}$ in $\mathbf{x}$.



(b) $\mathbf{y} = \texttt{MGN}(\mathbf{x}|\boldsymbol{\theta}_{mgn})$, where $\{\sigma_i(\,,\,)\}_{i=1}^{K}$ are convex, twice differentiable, nonnegative elementwise functions, and $\{s_i(\,.\,)\}_{i=1}^{K}$ are their respective derivatives.



(c) `MoGE` architecture uses `ICNN` and `MGN` alongside a gating network `Gate`, which is used to soft-select the better output from the two experts.

Figure 1: Schematic of the `MoGE` architecture, alongside its primary constituent elements `ICNN` and `MGN`.

A K-layer ICNN, denoted as $y = \mathbf{z}^K = \texttt{ICNN}(\mathbf{x}|\boldsymbol{\theta}_{icnn})$, uses the architecture

$$\mathbf{z}^{k+1} = \sigma_k\left(\mathbf{W}^{z,k}\mathbf{z}^k + \mathbf{W}^{x,k}\mathbf{x} + \mathbf{b}^k\right),\ k = 0, \ldots, K-1,$$

where $\mathbf{z}^i$ denotes the layer activations with $\mathbf{z}^0 \equiv \mathbf{0}$ and $\mathbf{W}^{z,0} \equiv \mathbf{0}$. σ_i are nonlinear activation functions and $\boldsymbol{\theta}_{icnn} = \{\mathbf{W}^{x,0:K-1}, \mathbf{W}^{z,1:K-1}, \mathbf{b}^{0:K-1}\}$ are the learnable parameters. If all weights $\mathbf{W}^{z,1:K-1}$ are nonnegative and all functions σ_i are convex and non-decreasing, then `ICNN` is convex in its input-to-output mapping [48, Prop. 1]. Consequently, $\nabla_{\mathbf{x}}\texttt{ICNN}(\mathbf{x}|\boldsymbol{\theta}_{icnn})$ can be used to learn the dual variables. On the other hand, `MGN` is a shallow architecture with K units given as

$$\mathbf{y} = \texttt{MGN}(\mathbf{x}|\boldsymbol{\theta}_{mgn}) = \mathbf{b}' + \mathbf{V}^\top\mathbf{V}\mathbf{x} + \sum_{k=1}^{K}\sigma_k(\mathbf{z}^k)\mathbf{A}^{k\top}s_k(\mathbf{z}^k),$$
$$\mathbf{z}^k = \mathbf{A}^k\mathbf{x} + \mathbf{b}^k\ \forall k \in [K],\ \ \boldsymbol{\theta}_{mgn} = \left\{\mathbf{A}^{1:K}, \mathbf{b}^{1:K}, \mathbf{V}, \mathbf{b}'\right\}.$$

Given convex, nonnegative, and twice differentiable elementwise activations σ_k with s_k being σ_k's elementwise derivative, it can be shown that the Jacobian of `MGN` is positive semi-definite [49, Prop. 2]. Therefore, `MGN` is suitable for directly learning the gradients of convex functions.

In Section 4, ablative testing will reveal limitations in the ability of either of these two models to effectively learn dual variables. To address this, we adopt a mixture-of-experts (MoE) architecture named `MoGE` (Mixture of Gradient Experts). This approach integrates the strengths of both models using an auxiliary gating network, aiming to enhance robustness and performance in learning dual

variables. MoE architectures, pioneered over three decades ago [50], are increasingly favored for their capability to leverage diverse model strengths effectively. Recent applications in large language models like Mixtral [51] underscore their suitability for complex tasks requiring nuanced learning and adaptation.

Letting $\boldsymbol{\chi}$ denote the elementwise sigmoid function, the overall output of `MoGE` is

$$\mathbf{y} = \boldsymbol{\chi}\left(\texttt{Gate}(\mathbf{x}|\boldsymbol{\theta}_{gate})\right)\texttt{ICNN}(\mathbf{x}|\boldsymbol{\theta}_{icnn}) + \left(\mathbf{1} - \boldsymbol{\chi}\left(\texttt{Gate}(\mathbf{x}|\boldsymbol{\theta}_{gate})\right)\right)\texttt{MGN}(\mathbf{x}|\boldsymbol{\theta}_{mgn}).$$

The training process involves pre-training `ICNN` and `MGN` for a certain number of steps, followed by training `Gate` on the outputs of the trained experts for the same number of epochs. We consider the mean-square error loss for the i^{th} data sample

$$l^{(i)} \triangleq \left\| [\texttt{MoGE}(\boldsymbol{\gamma}^{(i)}, \boldsymbol{\xi}^{(i)})]_{(\mathcal{I}_{\tilde{\boldsymbol{\lambda}}}, \mathcal{I}_{\tilde{\boldsymbol{\mu}}})} - \left(-\tilde{\boldsymbol{\lambda}}^{(i)*}, -\tilde{\boldsymbol{\mu}}^{(i)*}\right) \right\|_2^2, \tag{7}$$

where $(\mathcal{I}_{\tilde{\boldsymbol{\lambda}}}, \mathcal{I}_{\tilde{\boldsymbol{\mu}}})$ are indices corresponding to $\tilde{\boldsymbol{\lambda}}^*$ and $\tilde{\boldsymbol{\mu}}^*$. During the training of `Gate`, a weighing factor of $100N$ is used for loss indices that are ground truth binding, thereby allowing better learning of sparse binding constraints. The `Gate` network acts as a polling mechanism which soft-selects the response of the more accurate expert as the final response. We conclude this section by highlighting the necessity of using the mixture of two different models.

Remark 2 (Necessity of Using a Mixture-of-Experts Model)**.** *As will be seen in Algorithm 2 and Proposition 3, in order to achieve a C-OPF solution which is identical to the full problem, we must solve screened instances of the C-OPF at most $p+1$ times, where p denotes the number of false negatives, i.e. ground truth binding constraints which are classified as non-binding by the given CS method. In order to reduce solve time, p must be 0 on average, which necessitates that false negative rates must be brought close to zero. `ICNN`, while being competetive in terms of correctly classifying constraints, still suffers from unacceptably high rates of false negatives as seen in Section 4. On the other hand `MGN` tends to generally classify a large number of constraints as binding by the virtue of it being less performant due to its shallow architecture. This behavior of `MGN` can be used to counteract `ICNN`'s false negatives by combining the two using `Gate`, which is a strategy that has been previously used to extract enhanced performance out of a combination of biased experts [52].*

3.4. Dataset Generation and Training

The dataset generation and `MoGE` training process are highlighted in Algorithm 1. As seen in Table (1), the parameters $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ contain values such as real and reactive load demands and thermal limits. To enhance the robustness of CS to newly encountered values of such parameters, we first create dataset $\mathcal{D}_1$ by uniformly sampling $\boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ over given intervals. Each dimension of the parameter space is sampled independently. Then, problem (1) is solved with each of the sampled parameters. The resulting parameters $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ and optimal

Algorithm 1 Dataset Generation and Training `MoGE`

Input: Upper bound $(\bar{\boldsymbol{\gamma}}, \bar{\boldsymbol{\xi}})$ and lower bound $(\underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\xi}})$, number of initial samples K_1, number of convex hull samples K_2
Output: Trained `MoGE`

$\mathcal{D} = \{\}, \mathcal{D}_1 = \{\}, \mathcal{D}_2 = \{\}$
for $i = 1$ to K_1 **do** ▷ `Generate 1st dataset`
  Sample $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \sim \text{Unif}((\underline{\boldsymbol{\gamma}}, \underline{\boldsymbol{\xi}}), (\bar{\boldsymbol{\gamma}}, \bar{\boldsymbol{\xi}}))$
  Solve (1) with parameters $(\boldsymbol{\gamma}, \boldsymbol{\xi})$
  if (1) is infesible for $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ **then**, continue
  else
    Record inputs & outputs $\mathbf{d} \triangleq (\boldsymbol{\gamma}, \boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*)$
    $\mathcal{D}_1 = \mathcal{D}_1 \cup \{\mathbf{d}\}$
  end if
end for
for $i = 1$ to K_2 **do** ▷ `Generate 2nd dataset`
  Randomly generate a point $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \in \text{conv}\{(\boldsymbol{\gamma}^{(1)}, \boldsymbol{\xi}^{(1)}), \ldots, (\boldsymbol{\gamma}^{(|\mathcal{D}_1|)}, \boldsymbol{\xi}^{(|\mathcal{D}_1|)})\}$
  Solve (1) with the parameters $(\boldsymbol{\gamma}, \boldsymbol{\xi})$
  Record inputs & outputs $\mathbf{d} \triangleq (\boldsymbol{\gamma}, \boldsymbol{\xi}, \tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*)$
  $\mathcal{D}_2 = \mathcal{D}_2 \cup \{\mathbf{d}\}$
end for
$\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$
Train `MoGE` using minibatch training on losses $\{l^{(i)}\}_{i \in [|\mathcal{D}|]}$
return `MoGE`

Algorithm 2 Constraint Screening based Accelerated C-OPF

Input: Trained `MoGE`, feasible parameters $(\boldsymbol{\gamma}, \boldsymbol{\xi})$
Output: Solution $\mathbf{x}^*$ to C-OPF (1)

$\tilde{\boldsymbol{\lambda}}^* = [\texttt{MoGE}(\boldsymbol{\gamma}, \boldsymbol{\xi})]_{\mathcal{I}_{\tilde{\boldsymbol{\lambda}}}}$ ▷ `Predict optimal dual vars.`
$\mathcal{A}_{\text{bind}} = \{i : \tilde{\boldsymbol{\lambda}}_i^* \neq 0\}$ ▷ `Set of binding constr.`
$\mathcal{A}_{\text{viol}} = \varnothing$ ▷ `Set of violated constr.`
do
  Replace $\tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}$ in (1) with $[\tilde{\mathbf{g}}(\mathbf{x})]_{\mathcal{A}_{\text{bind}}} \preceq [\boldsymbol{\gamma}]_{\mathcal{A}_{\text{bind}}}$
  Solve the reduced version of (1)
  Save the optimal solution $\mathbf{x}^*$
  Set $\mathcal{A}_{\text{viol}} = \{i : \tilde{\mathbf{g}}_i(\mathbf{x}^*) > \boldsymbol{\gamma}_i\}$
  Update $\mathcal{A}_{\text{bind}} = \mathcal{A}_{\text{bind}} \cup \mathcal{A}_{\text{viol}}$
while $\mathcal{A}_{\text{viol}} \neq \varnothing$
return $\mathbf{x}^*$

dual variables $\tilde{\boldsymbol{\lambda}}^*, \tilde{\boldsymbol{\mu}}^*$ are all added to $\mathcal{D}_1$. If any parameter causes an infeasible problem, it is ignored. Three points are noteworthy. First, the sampling process can be replaced with the use of historic datasets from solving prior C-OPF prob-

lems. Second, employing the aforementioned sampling approach for generating the entire dataset could lead to numerous sampled parameters being infeasible, causing a significant waste of time. Therefore, we sample a distribution derived from the existing data to bootstrap the generation of additional feasible parameters (i.e. the set $\mathcal{D}_2$ in Algorithm 1). Third, the interior of the convex hull of $\mathcal{D}_1$ forms the open set $\Gamma \times \Xi$ discussed in theoretical analyses.

3.5. Accelerated C-OPF with CS

Following the training of `MoGE`, we use it to accelerate C-OPF as detailed in Algorithm 2. Given a new instance of $(\boldsymbol{\gamma}, \boldsymbol{\xi})$, `MoGE` predicts the value of $\tilde{\boldsymbol{\lambda}}^*$. Based on Assumption 1, non-zero elements of $\tilde{\boldsymbol{\lambda}}^*$ indicate the corresponding constraints must be binding. False positive mispredictions (classifying a non-binding constraint as binding) merely increase the reduced problem size without affecting the optimal solution. However, false negative mispredictions (classifying a binding constraint as non-binding) can alter the optimal solution. In such cases, Algorithm 2 adds the violated constraints to the list of binding constraints and re-solves C-OPF. The following result characterizes the number of re-solves as a function of model error.

Proposition 3. *The do-while loop in Algorithm 2 terminates after at most $p+1$ iterations, where p is the number of false negative constraints (constraints which bind but are classified by `MoGE` as non-binding in line 2 of Algorithm 2).*

Proposition 3 outlines the worst-case scenario and is proved in Appendix D. However, in practice, as shown in Section 4, our experiments consistently detect any violated constraints, if present, during the very first iteration of the do-while loop. Furthermore, re-solve is not even necessary for the majority of instances. These facts together ensure that the resulting C-OPF solve time can be significantly reduced. We also note that Algorithm 2 is compatible with CS methods beyond `MoGE`.

Remark 3 (Generality of Algorithm 2)**.** *In Algorithm 2, the sole role of `MoGE` is to generate the optimal dual variable $\tilde{\boldsymbol{\lambda}}^*$ for identifying binding constraints. Therefore, Algorithm 2 can be used with any alternative method capable of identifying binding constraints as a function of $(\boldsymbol{\gamma}, \boldsymbol{\xi})$. Furthermore, Prop. 3 continues to hold for CS methods other than `MoGE`, since it is only based on the properties of the C-OPF (*viz. *convexity, Prop. 1 and Assumption 1).*

We conclude this section with a result on the generalization capability of `MoGE` which extends [39, Theorem 6.4] to the `MoGE` architecture.

Theorem 2 (Generalization Error of `MoGE`)**.** *Let $\mathcal{A} \triangleq \{(\boldsymbol{\gamma}^{(1)}, \boldsymbol{\xi}^{(1)}), \cdots, (\boldsymbol{\gamma}^{(K)}, \boldsymbol{\xi}^{(K)})\} \subseteq \Gamma \times \Xi$ and $\nabla_{\boldsymbol{\gamma},\boldsymbol{\xi}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) = (-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*)$ for all $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \in \mathcal{A}$. Further, let $\mathcal{G}(\boldsymbol{\lambda}, \boldsymbol{\xi})$ and $\mathcal{H}(\boldsymbol{\lambda}, \boldsymbol{\xi})$, both defined on $\Gamma \times \Xi$, be parameterized monotone functions which are*

Table 2: Number of elements in different cases from the PGLIB and MATPOWER libraries, and solve times for the full problem.

Cases	**Buses**	**Lines**	**Generators**	**Transformers**	**Solve Time (Full Problem)**
PGLIB - Meshed					
`case118`	118	186	54	11	0.13s
`case1354`	1354	1991	260	240	4.81s
`case2312`	2312	3013	226	857	8.21s
`case4601`	4601	7199	408	2054	21.56s
`case10000`	10000	13139	2016	2374	49.52s
MATPOWER - Radial					
`case136`	136	135	1	0	2.74s
`case1197`	1197	1196	1	0	6.22s

convex gradients[2]*. Lastly, let* $\mathcal{T}(\boldsymbol{\lambda},\boldsymbol{\xi}) \in [0,1]$ *over* $\Gamma\times\Xi$ *and let* $\mathcal{Z} \triangleq \mathcal{T}\mathcal{G}+(1-\mathcal{T})\mathcal{H}$ *(arguments omitted). If* $\mathcal{G}$ *and* $\mathcal{H}$ *agree with* $\nabla\mathcal{V}$ *on all points in* $\mathcal{A}$*, then for all points* $(\boldsymbol{\gamma},\boldsymbol{\xi}) \in conv(\mathcal{A})$*,*

$$\nabla_{\boldsymbol{\gamma},\boldsymbol{\xi}}\mathcal{V}(\boldsymbol{\gamma},\boldsymbol{\xi}) = \mathcal{G}(\boldsymbol{\gamma},\boldsymbol{\xi}) = \mathcal{H}(\boldsymbol{\gamma},\boldsymbol{\xi}) = \mathcal{Z}(\boldsymbol{\gamma},\boldsymbol{\xi}) = (-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*).$$

The proof follows from standard properties of convex functions and is provided in Appendix E.

Remark 4 (`MoGE` Performance Improves with Dense Sampling)**.** *Theorem 2 colloquially states that if* `MoGE` *is trained to zero error over the training dataset* $\mathcal{D}$*, then it achieves zero test error over the convex hull of any subset of* $\mathcal{D}$ *such that* $\nabla\mathcal{V}$ *is constant over said subset. This observation motivates the idea that new samples from regions of* $\Gamma\times\Xi$ *with good sample coverage in the training dataset* $\mathcal{D}$ *enjoy more accurate outputs. This is because for cases where* $\mathcal{A}$ *belongs to a region of* $\Gamma\times\Xi$ *over which* $\mathcal{V}$ *is smooth and the points in* $\mathcal{A}$ *are clustered close together, the outputs of* $\nabla\mathcal{V}$ *are approximately equal for all points in* $\mathcal{A}$*.*

4. Simulations

We test the dataset generation Algorithm 1 and the accelerated C-OPF in Algorithm 2, focusing on the QC-OPF (2) and CDF-OPF models (A.1). We utilize the PGLIB and MATPOWER libraries for our test cases, with the former containing meshed transmission networks and the latter containing radial distribution networks. We select seven specific test cases which are detailed in Table 2. These test cases prescribe the grid topology, line and transformer parameters, nominal load demands, thermal limits, and PAD limits.

[2]It is possible for a function $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ to be monotone, yet for there to not exist a convex function ϕ such that $F \equiv \nabla\phi$ over $\text{dom}(\phi)$. `ICNN` and `MGN` avoid this problem: they are gradients of a convex field by construction.

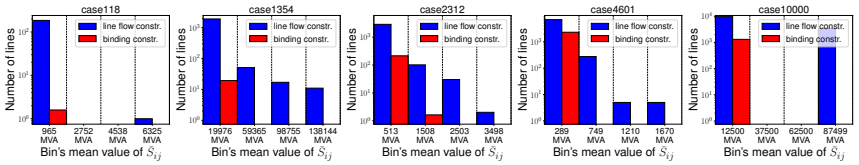


Figure 2: Histograms of thermal limits, along with the average number (across the test set and forward/reverse flows) of corresponding line flow constraints which bind for different cases in PGLIB. These constraints, which constitute $\tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}$, are candidates for removal via presolve.

Table 3: Performance evaluation of constraint classification approaches based on average solve time, training time, and fraction of re-solves for 100 test samples for QC-OPF. A negative reduction in solve time indicates an increase in solve time over the original problem.

Method of Constr. Removal	**Training Time**	**Confusion Matrix (Full Test Set)**				**Percent Reduction in Solve Time (100 samples)**	**Fraction Resolved (100 samples)**
		True Positive	**True Negative**	**False Positive**	**False Negative**		
case118							
Original problem	-	0.86%	99.14%	-	-	0.00%	-
MoGE	42.9275s	**0.86%**	94.35%	4.79%	**0.00%**	22.01%	**0.00**
ICNN	17.7126s	0.83%	95.68%	3.47%	0.02%	**22.16%**	**0.00**
MGN	2.1961s	0.83%	4.88%	94.26%	0.02%	-32.23%	**0.00**
Deep classifier	3.7815s	0.51%	**99.12%**	**0.02%**	0.34%	-27.61%	0.69
Ridge regression	3.3604s	0.52%	99.10%	0.04%	0.33	-32.46%	0.63
XGBoost	**1.5327s**	0.51%	99.12%	0.03%	0.34%	-38.39%	0.70
case1354							
Original problem	-	0.97%	99.03%	-	-	0.00%	-
MoGE	28.1495s	**0.97%**	97.24%	1.79%	**4e-5%**	**33.89%**	**0.00**
ICNN	12.8741s	0.97%	97.64%	1.39%	6e-4%	28.46%	0.09
MGN	11.4738s	0.95%	3.24%	95.79%	0.02%	-2.62%	0.10
Deep classifier	7.5701s	0.92%	**98.99%**	**0.04%**	0.05%	-3.57%	0.60
Ridge regression	**6.4307s**	0.92%	98.99%	0.04%	0.05%	-3.40%	0.60
XGBoost	33.3336s	0.90%	99.01%	0.02%	0.07%	-26.15%	0.97
case2312							
Original problem	-	7.14%	92.86%	-	-	0.00%	-
MoGE	49.6899s	**7.13%**	87.17%	5.69%	**3e-4%**	**25.38%**	**0.00**
ICNN	16.6255s	7.07%	89.91%	2.96%	0.07%	22.28%	0.06
MGN	25.6937s	7.12%	0.45%	92.42%	0.02%	14.99%	**0.00**
Deep classifier	**11.0422s**	6.92%	92.62%	0.25%	0.21%	-44.77%	1.00
Ridge regression	12.9950s	7.04%	91.57%	1.30%	0.09%	-39.94%	0.92
XGBoost	90.4948s	6.85%	**92.67%**	**0.19%**	0.29%	-45.04%	1.00
case4601							
Original problem	-	31.93%	68.07%	-	-	0.00%	-
MoGE	171.3697s	**31.93%**	58.49%	9.57%	**0.00%**	**20.15%**	**0.00**
ICNN	26.6654s	31.76%	59.28%	8.79%	0.17%	19.72%	**0.00**
MGN	101.7099s	31.87%	3.22%	64.85%	0.06%	18.02%	**0.00**
Deep classifier	**21.4779s**	31.91%	59.24%	8.83%	0.02%	18.99%	0.03
Ridge regression	40.9736s	31.91%	59.24%	8.83%	0.02%	19.22%	0.03
XGBoost	569.9688s	31.76%	**59.32%**	**8.75%**	0.17%	18.99%	0.03
case10000							
Original problem	-	9.94%	90.06%	-	-	0.00%	-
MoGE	580.2680s	**9.94%**	86.99%	3.07%	**0.00%**	**24.53%**	**0.00**
ICNN	**48.1584s**	8.87%	87.61%	2.44%	1.07%	20.21%	0.02
MGN	363.8684s	9.94%	26.54%	63.52%	0.00%	8.76%	**0.00**
Deep classifier	72.0453s	9.93%	**87.31%**	**2.75%**	0.01%	8.78%	0.17
Ridge regression	197.7062s	9.93%	87.31%	2.75%	0.01%	8.76%	0.17
XGBoost	-OOT-	-	-	-	-	-	-

Remark 5 (Dataset Generation)**.** *Variability in parameters* $(\boldsymbol{\gamma},\boldsymbol{\xi})$ *arises from perturbations around the prescribed nominal values of* P_i^d *and* Q_i^d *at each bus. Specifically, if the nominal values of the aforementioned quantities are* $\tilde{P}_i^d$ *and* $\tilde{Q}_i^d$*, then they are sampled i.i.d elemtwise in the boxes* $[0.25\tilde{P}_i^d, 1.75\tilde{P}_i^d]$ *and*

Table 4: Performance evaluation of constraint classification approaches based on average solve time, training time, and fraction of re-solves for 100 test samples for CDF-OPF.

Method of Constr. Removal	**Training Time**	**Confusion Matrix (Full Test Set)**				**Percent Reduction in Solve Time (100 samples)**	**Fraction Resolved (100 samples)**
		True Positive	**True Negative**	**False Positive**	**False Negative**		
case136							
Original problem	-	86.53%	13.47%	-	-	0.00%	-
MoGE	17.3123s	**86.53%**	9.85%	3.62%	**0.00%**	**9.87%**	**0.00**
ICNN	9.2234s	86.45%	10.72%	2.75%	0.08%	9.12%	0.04
MGN	3.1643s	86.50%	6.07%	7.40%	0.03%	7.22%	0.01
Deep classifier	4.2356s	85.91%	13.36%	0.11%	0.62%	-14.51%	0.86
Ridge regression	3.9744s	85.88%	13.34%	0.13%	0.65%	-13.95%	0.84
XGBoost	**2.0030s**	85.96%	**13.37%**	**0.10%**	0.57%	-17.80%	0.92
case1197							
Original problem	-	64.38%	35.62%	-	-	0.00%	-
MoGE	31.6432s	**64.38%**	33.02%	2.60%	**0.00%**	**26.41%**	**0.00**
ICNN	13.9122s	64.32%	33.52%	2.10%	0.06%	23.84%	0.05
MGN	17.8436s	64.34%	17.12%	18.50%	0.04%	12.36%	0.01
Deep classifier	**8.5220s**	64.08%	35.32%	0.30%	0.30%	-4.22%	0.40
Ridge regression	9.9355s	64.06%	35.31%	0.31%	0.32%	-5.91%	0.37
XGBoost	41.7673s	64.11%	**35.39%**	**0.23%**	0.27%	-2.13%	0.72

$[0.25\tilde{Q}_i^d, 1.75\tilde{Q}_i^d]$ *respectively. On the other hand, the nominal values of thermal limits* $\bar{S}_{ij}$ *are slightly perturbed per-instance in the range* $[(1-\epsilon)\bar{S}_{ij}, (1+\epsilon)\bar{S}_{ij}]$ *to ensure that the set* $\Gamma \times \Xi$ *has a nonzero measure (note that* $\bar{S}_{ij}$ *is replaced with* $\bar{S}_i$ *for the CDF-OPF model). We want to highlight that* $\Gamma \times \Xi$ *having a nonzero measure (i.e. no dimensions of* $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ *being constant across the dataset) is important for ensuring strong duality guarantees in Theorem 1 hold.*

The aforementioned sampling principle is used to generate 1000 values of $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ *to generate the dataset* $\mathcal{D}_1$*, with infeasible instances discarded (see Algorithm 1). Convex hull sampling is then used to create the augmented dataset* $\mathcal{D}_2$ *such that* $|\mathcal{D}| = |\mathcal{D}_1 \cup \mathcal{D}_2| = 5000$*. The first 4000 points in* $\mathcal{D}$ *are used for training, with the remaining points reserved for the test set.*

We use the Python implementation of the IPOPT solver [53], `cyipopt`, to solve optimization problems. This solver is augmented with customized code to ensure efficient solutions out of the box. `cyipopt` provides optimal primal-dual solutions, with the dual solutions used for subsequent simulation aspects.

4.1. Speeding up QC-OPF in `cyipopt`

As a Python implementation of IPOPT, `cyipopt` lacks the speed advantages of Julia's JIT-compiled IPOPT implementations. Nevertheless, we can achieve comparable speed improvements in Python by pre-computing coefficients as outlined herein. All implementations of IPOPT solve the problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & F(\mathbf{x}) \\ \text{s.t.} \quad & \underline{\mathbf{C}} \preceq \mathbf{C}(\mathbf{x}) \preceq \bar{\mathbf{C}}, \quad \underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}} \end{aligned}$$

wherein $\mathbf{C} : \mathbb{R}^n \mapsto \mathbb{R}^p$ are the lumped constraint functions, with equality constraints modeled by setting $\underline{\mathbf{C}}_i = \bar{\mathbf{C}}_i$. Given $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\alpha} \in \mathbb{R}^p$, and $\beta \in \mathbb{R}$, the user is required to provide routines that numerically compute the quantities

$F(\mathbf{x})$, $\mathbf{C}(\mathbf{x})$, $\nabla F(\mathbf{x})$, $\nabla \mathbf{C}(\mathbf{x})$, and $\mathbf{H}(\mathbf{x}, \boldsymbol{\alpha}, \beta) := \beta \nabla^2 F(\mathbf{x}) + \sum_{j=1}^{p} \alpha_j \nabla^2 \mathbf{C}_j(\mathbf{x})$. For QC-OPF, $\mathbf{H}$ is linear in $(\beta, \boldsymbol{\alpha})$ and independent of $\mathbf{x}$ while all other quantities are either linear or quadratic in $\mathbf{x}$. The same also holds for CDF-OPF. By pre-computing the coefficients of those terms, we achieve significant speedups since the computation of the above quantities reduces to sparse matrix-vector multiplications. For example, our implementation solves PGLIB `case10000` in 50 seconds without any constraint removal, compared to 67 seconds for the Julia-based IPOPT package *PowerModels.jl* [54], despite lacking Julia's JIT features.

Furthermore, quadratic thermal limits (for example, (2f) for QC-OPF) add significant computation overhead due to their contribution of linear terms to $\nabla \mathbf{C}(\mathbf{x})$ and constant terms to $\sum_{j=1}^{p} \alpha_j \nabla^2 \mathbf{C}_j(\mathbf{x})$. However, most of these constraints are not binding at the optimum, as shown in Figure 2. Therefore, reducing the solve time of any C-OPF involves accurately detecting and removing non-binding thermal limits, which are part of the constraints $\tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}$ in our model.

4.2. CS Classifiers

Our design of `MoGE` involves `ICNN` with $K = 5$ and layer sizes $(2N + 2E, N + E, 300, 150, 2N + 2E)$, and ELU ($\alpha = 1$) activations. `MGN` is chosen with $K = 2$, σ_k is chosen as ELU ($\alpha = 1$) and Softplus, and $\mathbf{V}^\top \mathbf{V}$ is chosen to have rank 200 for all cases. `Gate` is a 2-layer network with a hidden layer of size 100, and LeakyReLU activation. The Adam optimizer is used to train `MoGE` for 2000 steps with a batch size of 64. Once `MoGE` has been trained via Algorithm 1, 100 randomly selected examples from the test set are fed in Algorithm 2 to observe the acceleration of C-OPF. We train `MoGE` via PyTorch, which is carried out on a system with an Intel Core i9 processor, 64 GB of RAM, and two NVIDIA RTX 3090 GPUs.

We can compare the proposed `MoGE`-based approach with other benchmark classification methods:

- To conduct an ablation study of `MoGE`, we list the performance of standalone `ICNN` and `MGN`. The former is a state-of-the-art architecture for learning dual variables in convex problems; see [55].
- *Deep classifier* is an DNN, which takes $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ as input and produces $\tilde{L}$ scalars in the range $(0, 1)$ as output, representing the probability of each of the $\tilde{L}$ constraints binding [56]. We employ a 4-layer DNN with same layer sizes as the `MoGE`, and train it on dataset $\mathcal{D}$ using binary cross-entropy loss. Training is performed for 2,000 epochs using the Adam optimizer with a batch size of 64.
- *Ridge classifier* uses PyTorch to train a ridge regression classifier on $\mathcal{D}$. It first converts the targets (indicating binding and non-binding constraints) to $\{-1, 1\}$, then fits a linear model that maps the input parameters to these targets using ridge regression.

- *XGBoost classifier* involves training a gradient-boosted decision-tree classifier. The `XGBoost` package is used with a tree depth of 4, trained for 50 epochs at a learning rate of 0.1.

4.3. Numerical Results

All classifiers' predictions are fed into Algorithm 2. The comparative results are presented in Tables 3 and 4, including the training time, confusion matrix, total solve times, and the fraction of cases requiring resolution due to the detection of violated constraints.

Deep and ridge classifiers, along with `MGN`, train the fastest, whereas `MoGE` takes longer due to its MoE-based framework. However, `MoGE`'s training time scales well with the number of buses, unlike XGBoost, which becomes impractical for larger cases. For example, XGBoost training for `case10000` exceeds 12 hours and fails to complete.

The proposed `MoGE`-based CS method shows high true positives and low false negatives but has relatively lower true negatives and higher false positives. This outcome is acceptable because, in constraint classification, minimizing false negatives is critical to avoid constraint violations in the reduced problem, which would require re-solving and significantly increase the total runtime of Algorithm 2. `MoGE` achieves very low false negatives, helping to avoid re-solves and resulting in the lowest solve times in most cases. Additionally, `MoGE` achieves lower false negatives than `ICNN` and `MGN`, which is beneficial for larger cases. Interestingly, `MGN` tends to classify most constraints as binding, allowing `Gate` to switch to `MGN` if `ICNN` makes a false positive error.

The solve time advantage of `MoGE` ranges from 10% to 34%, which is significant when C-OPF needs to be solved repeatedly or when computational resources are limited.Furthermore, `MoGE` is more robust than `ICNN` in avoiding re-solves. Additionally, although the deep classifier and XGBoost offer good classification performance, they cannot match the efficiency of architectures like `MoGE` and `ICNN` designed to approximate convex functions and their gradients.

5. Conclusion

In this paper, we introduced a data-driven constraint screening approach to called `MoGE` to presolve convexified optimal power flow, utilizing neural network architecture that can learn convex gradients. The following analyses and experiments were carried out.

- We presented a general technique based on rank conditions to assess the suitability of a family of C-OPF models for constraint screening.
- We provided generalization results and a guarantee that our proposed framework leads to the exact solution as the original problem in finite time.
- We discussed methods for generating a dataset of C-OPF parameters, and augmenting the same.

- Numerical simulations illustrate that the proposed method significantly reduces the solve times for large-scale C-OPF problems.

Future research directions include the following.

- Extending `MoGE` to cases where the dual variables may be degenerate.
- Developing presolving techniques for power networks with changing topologies and discrete elements, e.g. switches.

Appendix A. CDF-OPF

CDF-OPF is a convex relaxation of ACOPF for low-voltage radial networks with a tree topology. Let $(\mathcal{N}_0, \mathcal{E})$ be the directed graph representing the power network, where $\mathcal{N}_0 \triangleq \{0\} \cup \mathcal{N}$ is the set of buses with 0 denoting a special leaf bus (often the point of common coupling). We assume that the branches in $\mathcal{E}$, denoted by (i, j) are oriented *away* from bus 0. Owing to the network's tree topology, each branch in $\mathcal{E}$ corresponds to a unique bus in $\mathcal{N}$, and as such we index a branch as j rather than its full description (i, j). Each bus $i \in \mathcal{N}$ is associated with indices $\mathcal{G}_i$ of generators attached to it, real and reactive demands P_i^d, Q_i^d, and minimum (maximum) voltage magnitude $\underline{V}_i$ ($\bar{V}_i$). We assume V_0 is fixed since it is determined by the upstream high-voltage transmission network. In addition, the real and reactive power outputs of the k^{th} generator are denoted by P_k^g and Q_k^g respectively. The set $\mathcal{G} \triangleq \cup_{i\in\mathcal{N}}\mathcal{G}_i$ collects all generation units. Each branch i is represented by its resistance r_i, reactance x_i, and thermal limit $\bar{S}_i$.

Let V_i denote the squared voltage magnitude and $p_i + \mathsf{j}q_i$ the complex injection at bus i. Furthermore, let $P_i + \mathsf{j}Q_i$ denote the complex power flow and ℓ_i the squared current magnitude in the direction of the branch i. Lastly, we use π_i to denote the *parent* of bus i, i.e. $(\pi_i, i) \in \mathcal{E}$, and the notation $i \to j$ to denote $(i, j) \in \mathcal{E}$. In this setup, the CDF-OPF formulation is given as follows.

$$\min\ f\left(\{P_k^g\}_{k\in\mathcal{G}}, \{Q_k^g\}_{k\in\mathcal{G}}, \{V_i\}_{i\in\mathcal{N}}, \{\ell_i\}_{i\in\mathcal{N}}\right), \tag{A.1a}$$

subject to:

$$\sum_{k:i\to k} P_k - p_i - P_i + r_i\ell_i = 0, \forall i \in \mathcal{N} \tag{A.1b}$$

$$\sum_{k:i\to k} Q_k - q_i - Q_i + x_i\ell_i = 0, \forall i \in \mathcal{N} \tag{A.1c}$$

$$V_{\pi_i} - V_i - 2r_iP_i - 2x_iQ_i + (r_i^2 + x_i^2)\ell_i = 0, \forall i \in \mathcal{N} \tag{A.1d}$$

$$P_i^2 + Q_i^2 \le V_{\pi_i}\ell_i, \forall i \in \mathcal{N} \tag{A.1e}$$

$$p_i - \left(\sum_{k\in\mathcal{G}_i} P_k^g\right) = -P_i^d, \forall i \in \mathcal{N} \tag{A.1f}$$

Table A.5: Representing CDF-OPF (A.1) as parametric convex optimization problem (1), where the box constraint (1d) corresponds to constraints (A.1j) and (A.1k).

Variables	Constraints	Parameters
$\mathbf{x}$: $\{P_k^g, Q_k^g\}_{k\in\mathcal{G}}$, $\{P_i, Q_i, \ell_i\}_{i\in\mathcal{N}}$, $\{p_i, q_i, v_i\}_{i\in\mathcal{N}}$	$g(\mathbf{x}), \tilde{g}(\mathbf{x})$: (A.1e), (A.1h)–(A.1i) $h(\mathbf{x}), \tilde{h}(\mathbf{x})$: (A.1b)–(A.1d), (A.1f)–(A.1g)	$\boldsymbol{\gamma}$: $\underline{p}_0, \bar{p}_0, \underline{q}_0, \bar{q}_0, \{\bar{S}_i^2\}_{i\in\mathcal{N}}$ $\boldsymbol{\xi}$: $\{P_i^d, Q_i^d\}_{i\in\mathcal{N}}$

$$q_i - \left(\sum_{k\in\mathcal{G}_i} Q_k^g\right) = -Q_i^d, \forall i \in \mathcal{N} \tag{A.1g}$$

$$\underline{p}_0 \le \sum_{k:0\to k} P_k \le \bar{p}_0, \quad \underline{q}_0 \le \sum_{k:0\to k} Q_k \le \bar{q}_0 \tag{A.1h}$$

$$P_i^2 + Q_i^2 \le \bar{S}_i^2, \forall i \in \mathcal{N} \tag{A.1i}$$

$$\underline{V}_i^2 \le V_i \le \bar{V}_i^2, \forall i \in \mathcal{N} \tag{A.1j}$$

$$\underline{P}_k^g \le P_k^g \le \bar{P}_k^g, \quad \underline{Q}_k^g \le Q_k^g \le \bar{Q}_k^g, \forall k \in \mathcal{G} \tag{A.1k}$$

In the above, the convex objective (A.1a) can model various objectives such as generation cost minimization, volt-VAR control, or minimization of I^2R losses in the network. Constraints (A.1b)–(A.1e) represent the convex relaxation of the DistFlow equations. (A.1f)–(A.1g) characterize per-bus injections, while (A.1h) bound the injections on bus 0. The thermal limits of each branch are specified in (A.1i). Lastly, (A.1j) places bounds on voltage magnitudes, while (A.1k) limits real and reactive generation quantities. CDF-OPF can be represented as a parameterized C-OPF (1) as shown in Table A.5.

Lemma 3. *If $V_0 > 0$ and the feasible set of* (A.1c)–(A.1e) *does not include $V_i = 0$ for any bus i, Lemma 2 holds for CDF-OPF.*

Proof. Let $N \triangleq |\mathcal{N}|$, and let $\tilde{\mathbf{A}} \in \mathbb{R}^{N\times N+1}$ be the branch-bus incidence matrix of $(\mathcal{N}_0, \mathcal{E})$. We can represent $\tilde{\mathbf{A}}$ as $\tilde{\mathbf{A}} = \begin{bmatrix}\mathbf{a}_0 & \mathbf{A}\end{bmatrix}$ wherein $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the invertible reduced branch-bus indicence matrix. Subsequently, we set up a few definitions. We let $\mathbf{F} \triangleq \mathbf{A}^{-1}$, $\mathbf{D}_r \triangleq \operatorname{diag}(\{r_i\}_{i\in\mathcal{N}})$ and $\mathbf{D}_x \triangleq \operatorname{diag}(\{x_i\}_{i\in\mathcal{N}})$, and correspondingly the matrices $\mathbf{R} \triangleq 2\mathbf{F}\mathbf{D}_r\mathbf{F}^\top$ and $\mathbf{X} \triangleq 2\mathbf{F}\mathbf{D}_x\mathbf{F}^\top$ are defined. In this setting, equations (A.1b)–(A.1e) can be written in matrix-vector notation as [57, (7)]

$$\mathbf{p} - \mathbf{A}^\top\mathbf{P} - \mathbf{D}_r\boldsymbol{\ell} = \mathbf{0} \tag{A.2a}$$

$$\mathbf{q} - \mathbf{A}^\top\mathbf{Q} - \mathbf{D}_x\boldsymbol{\ell} = \mathbf{0} \tag{A.2b}$$

$$\mathbf{A}\mathbf{v} + \mathbf{a}_0V_0 - 2\mathbf{D}_r\mathbf{P} - 2\mathbf{D}_x\mathbf{Q} + (\mathbf{D}_r^2 + \mathbf{D}_x^2)\boldsymbol{\ell} = \mathbf{0} \tag{A.2c}$$

$$\operatorname{diag}(\mathbf{P})\mathbf{P} + \operatorname{diag}(\mathbf{Q})\mathbf{Q} - \operatorname{diag}(\boldsymbol{\ell})\mathbf{v}_\pi \preceq \mathbf{0}, \tag{A.2d}$$

wherein $\mathbf{p} \triangleq \{p_i\}_{i\in\mathcal{N}}$, $\mathbf{q} \triangleq \{q_i\}_{i\in\mathcal{N}}$, $\mathbf{P} \triangleq \{P_i\}_{i\in\mathcal{N}}$, $\mathbf{Q} \triangleq \{Q_i\}_{i\in\mathcal{N}}$, and $\mathbf{v} \triangleq \{V_i\}_{i\in\mathcal{N}}$ gather the relevant variables in a vectorized form. Furthermore $\mathbf{v}_\pi \triangleq$

$\begin{bmatrix} V_{\pi_1} & V_{\pi_2} & \cdots & V_{\pi_N} \end{bmatrix}$ collects the parent bus' voltage of each bus. It remains to prove that the Jacobian of binding constraints in (A.2a)–(A.2d) are linearly independent with respect to any feasible $\mathbf{x}$ as defined in Table A.5.

To that end, we show linear independence of Jacobian rows with respect to a subset of variables (columns) given as

$$\tilde{\mathbf{x}} \triangleq \{\mathbf{p}, \mathbf{q}, \mathbf{v}, [\boldsymbol{\ell}]_{\mathcal{A}_\mathbf{x}}\},$$

wherein $\mathcal{A}_\mathbf{x} \subseteq [N]$ is the set of binding SOCP constraints (A.2d) for a given $\mathbf{x}$. Note that

$$\frac{\partial(\text{A.2a}), (\text{A.2b}), (\text{A.2c}), (\text{A.2d})}{\partial \tilde{\mathbf{x}}} = \text{blkdiag}\begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N & \mathbf{A} & [\text{diag}(\mathbf{v}_\pi)]_{\mathcal{A}_\mathbf{x}} \end{bmatrix},$$

where $[\text{diag}(\mathbf{v}_\pi)]_{\mathcal{A}_\mathbf{x}}$ selects the rows of $\text{diag}(\mathbf{v}_\pi)$ corresponding to the indices in $\mathcal{A}_\mathbf{x}$. By the invertibility of $\mathbf{A}$ and the nonzero nature of the vector $\mathbf{v}_\pi$ due to feasibility of $\mathbf{x}$, we conclude that the above partial jacobian has full row-rank. This concludes the proof. □

Lemma 3 shows that the convexified DistFlow model, under mild conditions (i.e. the voltage collapse mode $V_i = 0$ for any bus i is not feasible) satisfies the fixed LICQ conditions. Subsequently, it also satisfies the strong duality conditions over $\Gamma \times \Xi$ stated in Theorem 1.

Appendix B. Proof of Lemma 2

Let $\mathcal{A}_\mathbf{g}(\mathbf{x}) = \mathcal{A}_{\text{soc}}(\mathbf{x}) \cup \mathcal{A}_{\text{ang}}(\mathbf{x})$, where $\mathcal{A}_{\text{soc}}(\mathbf{x})$ is the set of SOC binding constraints in (2c). $\mathcal{A}_{\text{ang}}(\mathbf{x}) \triangleq \mathcal{A}^{low}_{\text{ang}}(\mathbf{x}) \cup \mathcal{A}^{up}_{\text{ang}}(\mathbf{x})$ is the set of angle binding constraints in (2d), which reflects the bindings at the PAD lower or upper limits.[3] Also, Let $l(m) : [E] \mapsto \mathcal{E}$ be the function mapping each index in $[E]$ to a unique line in $\mathcal{E}$. Define the partial set of variables $\mathbf{x}_s \triangleq \left\{P_{ij}, Q_{ij}, P_{ji}, Q_{ji}, W^{\text{R}}_{ij}, W^{\text{I}}_{ij}\right\}_{(i,j)\in\mathcal{E}}$. Let $\mathbf{J}_s$ be the submatrix of the Jacobian $\mathbf{J}$ formed by the columns corresponding to $\mathbf{x}_s$. We can compactly express it as a block upper triangular matrix of size $(4E + |\mathcal{A}_\mathbf{g}(\mathbf{x})|) \times 6E$, as follows:

$$\mathbf{J}_s = \begin{array}{c} \\ \text{(2b)} \\ \\ \text{(2c)} \\ \text{(2d)} \end{array} \begin{array}{c} \begin{array}{cccccc} P_{ij} & Q_{ij} & P_{ji} & Q_{ji} & W^{\text{R}}_{ij} & W^{\text{I}}_{ij} \end{array} \\ \left(\begin{array}{cccc|cc} & & & & & \\ & & \mathbf{I}_{4E} & & & * \\ & & & & & \\ \hline & & & & & \\ & & \mathbf{0} & & & \underbrace{\begin{bmatrix} \mathbf{S}^{\text{R}} & \mathbf{S}^{\text{I}} \\ \mathbf{A}^{\text{R}} & \mathbf{A}^{\text{I}} \end{bmatrix}}_{\mathbf{J}_g} \end{array}\right) \end{array}$$

where the four submatrices are given as:

$$\mathbf{S}^{\text{R}} = \left[2W^{\text{R}}_{l(m)}\mathbf{e}_m^\top : m \in \mathcal{A}_{\text{soc}}\right]$$

[3] Henceforth, we omit the argument $(\mathbf{x})$ for these sets to simplify notation.

$$\mathbf{S}^{\mathrm{I}} = \left[2W^{\mathrm{I}}_{l(m)}\mathbf{e}_m^\top : m \in \mathcal{A}_{\mathrm{soc}}\right]$$
$$\mathbf{A}^{\mathrm{R}} = \begin{bmatrix} \begin{cases} \tan(\underline{\theta}_{l(m)})\mathbf{e}_m^\top & \text{if } m \in \mathcal{A}^{low}_{\mathrm{ang}} \\ -\tan(\bar{\theta}_{l(m)})\mathbf{e}_m^\top & \text{if } m \in \mathcal{A}^{up}_{\mathrm{ang}} \end{cases} \end{bmatrix}$$
$$\mathbf{A}^{\mathrm{I}} = \begin{bmatrix} \begin{cases} -\mathbf{e}_m^\top & \text{if } m \in \mathcal{A}^{low}_{\mathrm{ang}} \\ \mathbf{e}_m^\top & \text{if } m \in \mathcal{A}^{up}_{\mathrm{ang}} \end{cases} \end{bmatrix}.$$

Note that these submatrices are constructed row by row using scaled canonical vectors $\{\mathbf{e}_m^\top\}$, with scaling factors determined by specific conditions. Furthermore, we have $W^{\mathrm{R}}_{l(m)} \neq 0, \forall m \in \mathcal{A}_{\mathrm{soc}}(\mathbf{x})$. Otherwise, $W^{\mathrm{I}}_{l(m)} = 0$ due to (2d), which implies $W_{ii} = 0$ for some i (cf. binding constraint (2c)). This would violate the box constraint (2h) with nonzero $\underline{V}_i^2$.

We will explore the special structure of $\mathbf{J}_g(\mathbf{x})$ to demonstrate its full rank for any $\mathbf{x} \in \mathcal{X}_{\mathrm{pf}}$, thereby establishing the full rank of $\mathbf{J}_s$ and $\mathbf{J}$ [58, Sec. 0.9.4]. First, if $\mathcal{A}_{\mathbf{g}} = \varnothing$, then $\mathbf{J}_s = \begin{bmatrix}\mathbf{I}_{4E} & *\end{bmatrix}$, which is full rank. Otherwise, depending on the binding possibilities of (2c) and (2d), we have four cases: i) $\mathcal{A}_{\mathrm{soc}} = \varnothing, \mathcal{A}_{\mathrm{ang}} \neq \varnothing$; ii) $\mathcal{A}_{\mathrm{soc}} \neq \varnothing, \mathcal{A}_{\mathrm{ang}} = \varnothing$; iii) $\mathcal{A}_{\mathrm{soc}} \neq \varnothing, \mathcal{A}_{\mathrm{ang}} \neq \varnothing, \mathcal{A}_{\mathrm{soc}} \cap \mathcal{A}_{\mathrm{ang}} = \varnothing$; and iv) $\mathcal{A}_{\mathrm{soc}} \cap \mathcal{A}_{\mathrm{ang}} \neq \varnothing$. In each of the first three cases, $\mathbf{J}_g$ is full rank as it is in *row echelon form (REF) without zero rows*, possibly after row permutations.

Finally, the most complex case-iv occurs when both the SoC constraint and one of the angle limit constraints are simultaneously binding for at least one line. Define $\rho_m \overset{\Delta}{=} \frac{-\tan(\underline{\theta}_{l(m)})}{2W^{\mathrm{R}}_{l(m)}}$, $\forall m \in \mathcal{A}_{\mathrm{soc}} \cap \mathcal{A}^{low}_{\mathrm{ang}}$. For each m, the two corresponding rows in $\mathbf{J}_g$ have only four nonzero elements. Applying the elementary row operation:

$$\begin{bmatrix} \cdots & 2W^{\mathrm{R}}_{l(m)} & \cdots & 2W^{\mathrm{I}}_{l(m)} & \cdots \\ \cdots & \tan(\underline{\theta}_{l(m)}) & \cdots & -1 & \cdots \end{bmatrix} \xrightarrow{r_2 + \rho_m \times r_1}$$
$$\begin{bmatrix} \cdots & 2W^{\mathrm{R}}_{l(m)} & \cdots & 2W^{\mathrm{I}}_{l(m)} & \cdots \\ \cdots & 0 & \cdots & p & \cdots \end{bmatrix},$$

we get the pivot $p = -(1+\tan^2(\underline{\theta}_{l(m)})) \neq 0$ for the second row. Similarly, define $\tilde{\rho}_m \overset{\Delta}{=} \frac{\tan(\bar{\theta}_{l(m)})}{2W^{\mathrm{R}}_{l(m)}}$, $\forall m \in \mathcal{A}_{\mathrm{soc}} \cap \mathcal{A}^{up}_{\mathrm{ang}}$, we get the pivot $\tilde{p} = 1 + \tan^2(\bar{\theta}_{l(m)}) \neq 0$ with the same row operation. By repeating the aforementioned operation for all $m \in \mathcal{A}_{\mathrm{soc}} \cap \mathcal{A}_{\mathrm{ang}}$, we convert $\mathbf{J}_g$ into *REF without zero rows*, confirming its full rank.

Appendix C. Proof of Theorem 1

Consider the following parametric (not necessarily convex) problem, where the parameter space Θ is an open, nonempty set with non-zero measure.

$$\begin{aligned} \mathcal{P}(\boldsymbol{\theta}) \overset{\Delta}{=} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) \preceq \mathbf{0}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned}$$

$$\tilde{\mathbf{g}}(\mathbf{x},\boldsymbol{\theta}) \preceq \mathbf{0}, \quad \tilde{\mathbf{h}}(\mathbf{x},\boldsymbol{\theta}) = \mathbf{0}.$$

Theorem (Genericity of LICQ [59])**.** *Consider the problem $\mathcal{P}(\boldsymbol{\theta})$ with $f, \mathbf{g}, \mathbf{h}, \tilde{\mathbf{g}}, \tilde{\mathbf{h}}$ in class C^r (i.e., r-times continuously differentiable) for all $\mathbf{x} \in \mathcal{X}_{\rm pf}$ and $\boldsymbol{\theta} \in \Theta$ with $r > \max(0, n - M - \tilde{M})$ and assume LICQ holds for all $\mathbf{x} \in \mathcal{X}_{\rm pf}$. If the map $\boldsymbol{\theta} \mapsto (\tilde{\mathbf{g}}(\mathbf{x},\boldsymbol{\theta}), \tilde{\mathbf{h}}(\mathbf{x},\boldsymbol{\theta}))$ has rank $\tilde{M} + \tilde{L}$ for every $\mathbf{x} \in \mathcal{X}_{\rm pf}$ and $\boldsymbol{\theta} \in \Theta$, then LICQ holds for almost every $\boldsymbol{\theta} \in \Theta$ and for every feasible point of $\mathcal{P}(\boldsymbol{\theta})$.*

To prove Theorem 1, we will demonstrate that QC-OPF satisfies all conditions stated in the theorem above. That is, C1) all functions are in class C^r; C2) the parameter-to-constraint map is full rank; and C3) LICQ holds for all $\mathbf{x} \in \mathcal{X}_{\rm pf}$, which is already established in Lemma 2. First, since all functions in QC-OPF are either affine or quadratic, they are infinitely differentiable, thereby satisfying condition C1). Second, let us consider embedding the box constraint $\underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}}$ into $\tilde{\mathbf{g}}$. The parameter-to-constraint map of C-OPF is $(\boldsymbol{\gamma}, \boldsymbol{\xi}) \mapsto (\tilde{\mathbf{g}}(\mathbf{x}) - \boldsymbol{\gamma}, \tilde{\mathbf{h}}(\mathbf{x}) - \boldsymbol{\xi})$. It is evident that the mapping's Jacobian is the diagonal sign matrix (± 1 on the diagonal). Thus, the map is full-rank. Finally, it is known that LICQ implies *Mangasarian-Fromovitz constraint qualification (MFCQ)*, which is equivalent to Slater's condition for convex problems [60, pp. 45, 160]. This result ensures strong duality and hence completes the proof.

Appendix D. Proof of Proposition 3

For brevity, we define the set

$$C(\boldsymbol{\xi}) \triangleq \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \preceq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}, \tilde{\mathbf{h}}(\mathbf{x}) = \boldsymbol{\xi}, \underline{\mathbf{x}} \preceq \mathbf{x} \preceq \bar{\mathbf{x}} \right\},$$

which allows us to restate (1) as

$$\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) = \min_{\mathbf{x} \in C(\boldsymbol{\xi})} \quad f(\mathbf{x}), \quad \text{s.t.} \quad \tilde{\mathbf{g}}(\mathbf{x}) \preceq \boldsymbol{\gamma}, \tag{D.1}$$

with its optimizer denoted by $\mathbf{x}^*$. Consider the following lemma.

Lemma 4. *For problem D.1, suppose it is known* a priori *that $\tilde{\boldsymbol{\lambda}}_i^* > 0$ for $i \in S \subseteq [\tilde{L}]$. Then, the optimizer $\mathbf{x}^\epsilon$ of the perturbed problem $\mathcal{V}(\boldsymbol{\gamma} + \epsilon\Delta, \boldsymbol{\xi})$, where*

$$\Delta \triangleq \begin{cases} 1, & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases}$$

violates at least one constraint in $[\tilde{\mathbf{g}}(\mathbf{x})]_S \preceq [\boldsymbol{\gamma}]_S$, for any $\epsilon > 0$.

Proof. Since it is known *a priori* that $[\tilde{\boldsymbol{\lambda}}^*]_S = -[\nabla_{\boldsymbol{\gamma}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi})]_S \succ \mathbf{0}$, and from the fact that $\nabla_{\boldsymbol{\gamma}} \mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) \preceq \mathbf{0}$ for any feasible $(\boldsymbol{\gamma}, \boldsymbol{\xi})$ due to Prop. 1 and dual variable properties, it follows that $\mathcal{V}(\boldsymbol{\gamma}, \boldsymbol{\xi}) > \mathcal{V}(\boldsymbol{\gamma} + \epsilon\Delta, \boldsymbol{\xi})$ for all $\epsilon > 0$. Consequently, we have $f(\mathbf{x}^*) > f(\mathbf{x}^\epsilon)$. To prove by contradiction, assume that $[\tilde{\mathbf{g}}(\mathbf{x}^\epsilon)]_S \preceq [\boldsymbol{\gamma}]_S$, i.e. the perturbed solution $\mathbf{x}^\epsilon$ does not violate any constraints in (D.1). However, this implies that $\mathbf{x}^\epsilon$ is the optimizer to (D.1) rather than $\mathbf{x}^*$. This leads to a contradiction, showing that the assumption is false. □

We set S to be the set of false negative constraints. Lemma 4 guarantees that at least one constraint violation will be detected in every iteration of the do-while loop and added to $\mathcal{A}_{\text{bind}}$. Thus, if there are p false negatives initially, the do-while loop must run no more than $p+1$ iterations.

Appendix E. Proof of Theorem 2

We start this proof by observing certain properties of convex functions and their gradients. Consider a monotone function $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ such that $f = \nabla_{\mathbf{x}}\Phi(\mathbf{x})$ for some convex $\Phi : \mathcal{X} \mapsto \mathbb{R}$ over $\mathcal{X}$. Let the discrete set $\bar{\mathcal{X}} \triangleq \{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(K)}\} \subset \mathcal{X}$ be such that $\mathbf{x}^{(i)} \in \text{dom}(\Phi)$ and $f(\mathbf{x}^{(i)}) = \mathbf{g}$ for all $i \in [K]$. Let an arbitrary point $\tilde{\mathbf{x}} \in \text{conv}\left(\bar{\mathcal{X}}\right)$ be represented as $\tilde{\mathbf{x}} = \sum_{i\in[K]} \boldsymbol{\alpha}_i \mathbf{x}^{(i)}$, where $\boldsymbol{\alpha} \succeq \mathbf{0}$ an $\mathbf{1}^\top\boldsymbol{\alpha} = 1$. It follows from the convexity of Φ that

$$\Phi(\tilde{\mathbf{x}}) \geq \Phi(\mathbf{x}^{(i)}) + \left(\nabla\Phi(\mathbf{x}^{(i)})\right)^\top \left(\tilde{\mathbf{x}} - \mathbf{x}^{(i)}\right). \tag{E.1}$$

Multiplying (E.1) with α_i and adding the inequalities over all i, we get

$$\Phi\left(\sum_{i\in[K]} \boldsymbol{\alpha}_i \mathbf{x}^{(i)}\right) \geq \sum_{i\in[K]} \boldsymbol{\alpha}_i \Phi(\mathbf{x}^{(i)}). \tag{E.2}$$

On the other hand, Jensen's inequality provides

$$\Phi\left(\sum_{i\in[K]} \boldsymbol{\alpha}_i \mathbf{x}^{(i)}\right) \leq \sum_{i\in[K]} \boldsymbol{\alpha}_i \Phi(\mathbf{x}^{(i)}). \tag{E.3}$$

(E.2) and (E.3) imply that equality holds both for any value of $\boldsymbol{\alpha}$ in the standard simplex. Taking the gradient of both sides of this equality with respect to $\mathbf{x}$, we obtain $f(\mathbf{x}) = \sum_{i\in[K]} \boldsymbol{\alpha}_i f(\mathbf{x}^{(i)}) = \mathbf{g}$ for all $\mathbf{x} \in \text{conv}\left(\bar{\mathcal{X}}\right)$.

In the context of the the present Theorem, we replace $\mathcal{X}$ with $\mathcal{A}$ and f with $\nabla\mathcal{V}$, $\mathcal{G}$, and $\mathcal{H}$ to observe that all three are identically equal to $(-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*)$ over $\text{conv}(\mathcal{A})$. Correspondingly the convex combination $\mathcal{Z}$ is also identically equal to $(-\tilde{\boldsymbol{\lambda}}^*, -\tilde{\boldsymbol{\mu}}^*)$ over $\text{conv}(\mathcal{A})$.

References

[1] M. B. Cain, R. P. O'neill, A. Castillo, et al., History of optimal power flow and formulations, Federal Energy Regulatory Commission 1 (2012) 1–36.

[2] Z. Qiu, G. Deconinck, R. Belmans, A literature survey of optimal power flow problems in the electricity market context, in: 2009 IEEE/PES Power Systems Conference and Exposition, 2009, pp. 1–6. doi:10.1109/PSCE.2009.4840099.

[3] S. Bose, Y. Zhang, Load restoration in islanded microgrids: Formulation and solution strategies, IEEE Transactions on Control of Network Systems 11 (3) (2024) 1345–1357. doi:10.1109/TCNS.2023.3337710.

[4] A. J. Wood, B. F. Wollenberg, G. B. Sheblé, Power generation, operation, and control, John Wiley & Sons, 2013.

[5] R. A. Jabr, Radial distribution load flow using conic programming, IEEE Trans. Power Syst. 21 (3) (2006) 1458–1459.

[6] M. Farivar, C. R. Clarke, S. H. Low, K. M. Chandy, Inverter VAR control for distribution systems with renewables, in: 2011 IEEE Intl Conf. on Smart Grid Commun., pp. 457–462.

[7] M. Farivar, S. H. Low, Branch flow model: Relaxations and convexification—part I, IEEE Trans. Power Syst. 28 (3) (2013) 2554–2564.

[8] X. Bai, H. Wei, K. Fujisawa, Y. Wang, Semidefinite programming for optimal power flow problems, International Journal of Electrical Power & Energy Systems 30 (6-7) (2008) 383–392.

[9] R. Madani, S. Sojoudi, J. Lavaei, Convex relaxation for optimal power flow problem: Mesh networks, IEEE Trans. Power Syst. 30 (1) (2015) 199–211. doi:10.1109/TPWRS.2014.2322051.

[10] J. Lavaei, S. H. Low, Zero duality gap in optimal power flow problem, IEEE Trans. Power Syst. 27 (1) (2012) 92–107. doi:10.1109/TPWRS.2011.2160974.

[11] S. Bose, S. H. Low, T. Teeraratkul, B. Hassibi, Equivalent relaxations of optimal power flow, IEEE Transactions on Automatic Control 60 (3) (2015) 729–742. doi:10.1109/TAC.2014.2357112.

[12] C. Coffrin, H. L. Hijazi, P. Van Hentenryck, The QC relaxation: A theoretical and computational study on optimal power flow, IEEE Transactions on Power Systems 31 (4) (2016) 3008–3018. doi:10.1109/TPWRS.2015.2463111.

[13] B. Kocuk, S. S. Dey, X. A. Sun, Strong SOCP relaxations for the optimal power flow problem, Operations Research 64 (6) (2016) 1177–1196.

[14] D. K. Molzahn, I. A. Hiskens, A survey of relaxations and approximations of the power flow equations, Foundations and Trends® in Electric Energy Systems 4 (1-2) (2019) 1–221.

[15] Z. Yang, H. Zhong, Q. Xia, C. Kang, Solving opf using linear approximations: fundamental analysis and numerical demonstration, IET Generation, Transmission & Distribution 11 (17) (2017) 4115–4125.

[16] S. H. Low, Convex relaxation of optimal power flow—part ii: Exactness, IEEE Transactions on Control of Network Systems 1 (2) (2014) 177–189. doi:10.1109/TCNS.2014.2323634.

[17] J. Lavaei, S. H. Low, Zero duality gap in optimal power flow problem, IEEE Transactions on Power Systems 27 (1) (2012) 92–107. doi:10.1109/TPWRS.2011.2160974.

[18] S. Sharif, J. Taylor, Minlp formulation of optimal reactive power flow, in: Proceedings of the 1997 American Control Conference (Cat. No.97CH36041), Vol. 3, 1997, pp. 1974–1978 vol.3. doi:10.1109/ACC.1997.611033.

[19] W. Huang, M. Chen, S. H. Low, Unsupervised learning for solving ac optimal power flows: Design, analysis, and experiment, IEEE Transactions on Power Systems 39 (6) (2024) 7102–7114. doi:10.1109/TPWRS.2024.3373399.

[20] P. L. Donti, D. Rolnick, J. Z. Kolter, Dc3: A learning method for optimization with hard constraints, arXiv preprint arXiv:2104.12225 (2021).

[21] K. Chen, S. Bose, Y. Zhang, Unsupervised deep learning for ac optimal power flow via lagrangian duality, in: GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 2022, pp. 5305–5310. doi:10.1109/GLOBECOM48099.2022.10000707.

[22] K. Chen, S. Bose, Y. Zhang, Physics-informed gradient estimation for accelerating deep learning-based ac-opf, IEEE Transactions on Industrial Informatics (2025) 1–12doi:10.1109/TII.2025.3545080.

[23] D. Owerko, F. Gama, A. Ribeiro, Optimal power flow using graph neural networks, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 5930–5934. doi:10.1109/ICASSP40776.2020.9053140.

[24] E. Naderi, L. Mirzaei, M. Pourakbari-Kasmaei, F. V. Cerna, M. Lehtonen, Optimization of active power dispatch considering unified power flow controller: application of evolutionary algorithms in a fuzzy framework, Evolutionary Intelligence 17 (3) (2024) 1357–1387.

[25] E. Naderi, L. Mirzaei, J. P. Trimble, D. A. Cantrell, Multi-objective optimal power flow incorporating flexible alternating current transmission systems: Application of a wavelet-oriented evolutionary algorithm, Electric Power Components and Systems 52 (5) (2024) 766–795. arXiv:https://doi.org/10.1080/15325008.2023.2234378, doi:10.1080/15325008.2023.2234378.
URL https://doi.org/10.1080/15325008.2023.2234378

[26] C. Roa-Sepulveda, B. Pavez-Lazo, A solution to the optimal power flow using simulated annealing, International Journal of Electrical Power & Energy Systems 25 (1) (2003) 47–57. doi:https://doi.org/10.1016/S0142-0615(02)00020-0.
URL https://www.sciencedirect.com/science/article/pii/S0142061502000200

[27] M. Gomez-Gonzalez, A. López, F. Jurado, Optimization of distributed generation systems using a new discrete pso and opf, Electric Power Systems Research 84 (1) (2012) 174–180. doi:https://doi.org/10.1016/j.epsr.2011.11.016.
URL https://www.sciencedirect.com/science/article/pii/S0378779611002835

[28] J. Telgen, Identifying redundant constraints and implicit equalities in systems of linear constraints, Management Science 29 (10) (1983) 1209–1222.

[29] G. L. Thompson, F. M. Tonge, S. Zionts, Techniques for removing nonbinding constraints and extraneous variables from linear programming problems, Manag. Sci. 12 (7) (1966) 588–608.

[30] Z. J. Zhang, P. T. Mana, D. Yan, Y. Sun, D. K. Molzahn, Study of active line flow constraints in DC optimal power flow problems, in: 2020 SoutheastCon, 2020, pp. 1–8. doi:10.1109/SoutheastCon44009.2020.9249749.

[31] B. Hua, Z. Bie, C. Liu, G. Li, X. Wang, Eliminating redundant line flow constraints in composite system reliability evaluation, IEEE Trans. Power Syst. 28 (3) (2013) 3490–3498. doi:10.1109/TPWRS.2013.2248762.

[32] R. Weinhold, R. Mieth, Fast security-constrained optimal power flow through low-impact and redundancy screening, IEEE Trans. Power Syst. 35 (6) (2020) 4574–4584. doi:10.1109/TPWRS.2020.2994764.

[33] S. Zhang, H. Ye, F. Wang, Y. Chen, S. Rose, Y. Ma, Data-aided offline and online screening for security constraint, IEEE Trans. Power Syst. 36 (3) (2021) 2614–2622. doi:10.1109/TPWRS.2020.3040222.

[34] A. Porras, S. Pineda, J. M. Morales, A. Jiménez-Cordero, Cost-driven screening of network constraints for the unit commitment problem, IEEE Trans. Power Syst. 38 (1) (2023) 42–51. doi:10.1109/TPWRS.2022.3160016.

[35] D. Deka, S. Misra, Learning for DC-OPF: Classifying active sets using neural nets, in: 2019 IEEE Milan PowerTech, 2019, pp. 1–6. doi:10.1109/PTC.2019.8810819.

[36] F. Hasan, A. Kargarian, Topology-aware learning assisted branch and ramp constraints screening for dynamic economic dispatch, IEEE Trans. Power Syst. 37 (5) (2022) 3495–3505. doi:10.1109/TPWRS.2022.3142957.

[37] F. Fioretto, T. W. Mak, P. Van Hentenryck, Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods, Proceedings of the AAAI Conference on Artificial Intelligence 34 (01) (2020) 630–637.

[38] M. K. Singh, V. Kekatos, G. B. Giannakis, Learning to solve the AC-OPF using sensitivity-informed deep neural networks, IEEE Trans. Power Syst. 37 (4) (2022) 2833–2846. doi:10.1109/TPWRS.2021.3127189.

[39] L. Zhang, Y. Chen, B. Zhang, A convex neural network solver for DCOPF with generalization guarantees, IEEE Trans. on Control of Network Syst. 9 (2) (2022) 719–730. doi:10.1109/TCNS.2021.3124283.

[40] R. Nellikkath, S. Chatzivasileiadis, Physics-informed neural networks for AC optimal power flow, Electric Power Systems Research 212 (2022) 108412.

[41] S. Babaeinejadsarookolaee, A. Birchfield, R. Christie, C. Coffrin, et al., The power grid library for benchmarking AC optimal power flow algorithms, arXiv preprint arXiv:1908.02788 (2019).

[42] R. D. Zimmerman, C. E. Murillo-Sánchez, R. J. Thomas, Matpower: Steady-state operations, planning, and analysis tools for power systems research and education, IEEE Trans. Power Syst. 26 (1) (2011) 12–19. doi:10.1109/TPWRS.2010.2051168.

[43] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, 2004.

[44] J. Matouek, B. Gärtner, Understanding and Using Linear Programming, Springer, 2006.

[45] G. Wachsmuth, On LICQ and the uniqueness of Lagrange multipliers, Operations Research Letters 41 (1) (2013) 78–80.

[46] M. Madrigal, V. Quintana, Degeneracy and duality gap on optimization-based electricity auctions, in: DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No.00EX382), 2000, pp. 332–337. doi:10.1109/DRPT.2000.855686.

[47] G. Pataki, L. Tunçel, On the generic properties of convex optimization problems in conic form, Mathematical Programming 89 (3) (2001) 449–457.

[48] B. Amos, L. Xu, J. Z. Kolter, Input convex neural networks, in: D. Precup, Y. W. Teh (Eds.), Proceedings of the 34th Intl. Conf. on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 146–155.

[49] S. Chaudhari, S. Pranav, J. M. Moura, Learning gradients of convex functions with monotone gradient networks, in: 2023 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5. doi:10.1109/ICASSP49357.2023.10097266.

[50] R. Jacobs, M. Jordan, S. Nowlan, G. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87. doi:10.1162/neco.1991.3.1.79.

[51] A. Jiang, A. Sablayrolles, A. Roux, A. Mensch, et al., Mixtral of experts, arXiv preprint arXiv:2401.04088 (2024).

[52] A. Abbas, Y. Andreopoulos, Biased mixtures of experts: Enabling computer vision inference under data transfer limitations, IEEE Transactions on Image Processing 29 (2020) 7656–7667. doi:10.1109/TIP.2020.3005508.

[53] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (2006) 25–57.

[54] C. Coffrin, R. Bent, K. Sundar, Y. Ng, M. Lubin, Powermodels. jl: An open-source framework for exploring power flow formulations, in: 2018 Power Syst. Computation Conf. (PSCC), IEEE, 2018, pp. 1–8.

[55] Y. Chen, L. Zhang, B. Zhang, Learning to solve DCOPF: A duality approach, Electric Power Systems Research 213 (2022) 108595.

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

[57] D. Deka, V. Kekatos, G. Cavraro, Learning distribution grid topologies: A tutorial, IEEE Transactions on Smart Grid 15 (1) (2024) 999–1013. doi:10.1109/TSG.2023.3271902.

[58] R. Horn, C. Johnson, Matrix analysis, 2nd Edition, Cambridge University Press, 2012.

[59] A. Hauswirth, S. Bolognani, G. Hug, F. Dörfler, Generic existence of unique Lagrange multipliers in AC optimal power flow, IEEE Control Systems Letters 2 (4) (2018) 791–796. doi:10.1109/LCSYS.2018.2849598.

[60] J. Borwein, A. Lewis, Convex Analysis and Nonlinear Optimization: Theory and Examples, 2nd Edition, Springer, 2006.