

Expediting AC Contingency Analysis using a Basecase Machine Learning Model

Md Obaidur Rahman, *Student Member, IEEE*, and Vassilis Kekatos, *Senior Member, IEEE*

Abstract—Grid planning and operation under increasingly variable operating conditions require fast and accurate AC power flow (AC-PF) contingency analysis for numerous line outages. While ML models can accelerate these computations, existing approaches often require either training separate models for different contingencies or a single model using data from multiple topologies. Both approaches incur substantial offline data generation and training costs. To address this gap, this work proposes a fixed-point framework that reuses a single ML model trained exclusively on basecase topology data to predict post-contingency AC-PF states under any non-critical single-line outage. We analyze the proposed method's convergence under the DC model approximation and derive its convergence rate in terms of network parameters. As a side result, we show that power transfer distribution factors (PTDFs) for non-critical lines have magnitudes less than one. We further characterize the ML training region to accommodate PF specifications encountered during fixed-point iterations and derive bounds on ML prediction errors for post-contingency state estimates. Numerical tests on a 6,717-bus synthetic Texas system demonstrate convergence across all non-critical single-line outages, with prediction errors remaining close to those of basecase. The proposed framework offers a favorable tradeoff between the speed of DC solvers and Newton-Raphson accuracy.

Index Terms—Contingency analysis, PTDF, fixed-point iteration, basecase topology, deep neural networks, line outage.

I. INTRODUCTION

Contingency analysis is a fundamental yet computationally formidable task in power system operation and planning. It assesses the system's ability to withstand credible component outages that may lead to line overloads, voltage instability, or cascading failures. A power system is deemed $N - k$ secure if it can withstand any credible failure of k major components, such as transmission lines, transformers, or generators [1]. In its preventive form, contingency analysis ensures that the system remains secure under any contingency without requiring any adjustments. In its corrective form, the analysis provides an action plan for each contingency, assuming the operator has sufficient time to respond. The rapid integration of data center loads and inverter-interfaced distributed energy resources intensifies the need for increasingly refined AC contingency analyses. This work shows how a system operator can repurpose a single ML model trained to predict basecase AC-PF states to compute $N - 1$ post-contingency AC-PF states.

To expedite contingency analysis, substantial work has studied sensitivity-based approximations of post-contingency power flows. Power transfer distribution factors (PTDFs) and line outage distribution factors (LODFs) provide closed-form estimates of active power flows under the DC-PF model [2]. Subsequent studies developed more efficient procedures for evaluating these factors. For example, [3], and [4] derived

direct LODF formulations and methods that exploit the sparsity of grid topologies. Reference [5] leverages the sparse representation of line outages to detect topology changes. These methods are well suited for rapid contingency screening, but their reliance on the DC-PF approximation precludes an accurate representation of voltage magnitudes, network losses, and apparent power limits.

To estimate these quantities while preserving computational efficiency, researchers have developed linearized approximations of the AC-PF equations. Researchers have used AC sensitivity factors for security assessment under uncertain operating conditions [6]. Reference [7] derived linear approximations of the AC-PF equations in rectangular coordinates, and [8] developed a linear-programming approximation of AC power flow. Although such approaches are more accurate than the DC-PF model, their performance depends heavily on the linearization point and the reference topology. To avoid linearization inaccuracies, reference [9] suggested warm-starting the post-contingency AC-PF solver from the pre-contingency state. In fact, the inverse PF Jacobian required in the first Newton-Raphson (NR) iteration for the post-contingency state can be efficiently computed from the pre-contingency inverse PF Jacobian [10]. Holomorphic embedding has also been applied to AC contingency analysis to improve convergence near stressed operating conditions where NR methods may fail to converge [11]. Although these methods improve the efficiency or robustness of post-contingency AC-PF solvers, contingency analysis still requires repeated solutions across many loading and outage scenarios. This computational burden has motivated data-driven surrogate models.

Among these data-driven models, DNN-based approaches have been developed for learning optimal power flow (OPF) solutions [12], [13], [14], corrective actions [15], state estimation and in other grid design and security assessment tasks [16], [17], [18], [19]. Physics-guided NNs incorporate the AC-PF equations into the training process to improve the physical consistency of the predicted states [20], [21]. Learning-based methods have also been suggested specifically for contingency-constrained problems. Reference [22] accelerates $N - 1$ contingency screening using a deep convolutional neural network to find post-contingency AC-PF solutions. However, it requires training on numerous scenarios and topologies.

To reduce dependence on topology-specific training, researchers have explored topology-aware learning architectures. Graph neural networks (GNNs) address this issue by incorporating the power system topology directly into the learning architecture and have been applied to contingency prediction [23]. More recently, [24] developed a universal GNN for power system state estimation across different topologies.

Nonetheless, training across multiple scenarios and topologies remains a major computational burden. Reference [25] uses an input-convex NN to classify whether a power system is $N-k$ secure under the DC-PF model, yet that is a relatively lighter task than predicting the post-contingency states.

Can a model trained solely to solve the AC-PF problem under the basecase topology be directly reused to predict post-contingency states? This work answers this question affirmatively and makes four contributions:

- c1) We propose predicting post-contingency AC-PF states via a fixed-point iteration that reuses a single ML model trained exclusively on the basecase topology (Section II).
- c2) To study the convergence of the proposed scheme, we analyze the fixed-point iterations derived from the DC-PF model and establish convergence for all non-critical single-line outages (Section III). As a byproduct of broader interest, we establish bounds on PTDs.
- c3) We characterize the expanded set of AC-PF specifications needed to train the base-case ML model so that the fixed-point iteration operates properly (Section IV).
- c4) We analyze how ML prediction errors propagate through the fixed-point iteration to the final post-contingency state estimate (Section V).

We numerically tested the proposed method on the synthetic Texas 6,717-bus system using a DNN trained solely using basecase data. The proposed fixed-point iteration converges for all non-critical line contingencies. The empirical convergence rate and prediction accuracy corroborate our analytical findings. Overall, the method achieves a favorable speed-accuracy tradeoff, substantially reducing runtime relative to NR while maintaining higher accuracy over the DC-based solver.

The questions addressed under c2) and c3) were also studied in the conference precursor of this work [26]. In [26], we relied on semidefinite programming (SDP) formulations to bound the sensitivity of the basecase DNN with respect to its AC-PF specification inputs. Although this SDP-based analysis bounds the exact DNN fixed-point iteration, it is computationally demanding, limited to DNNs, and can yield loose sensitivity bounds that make the resulting guarantees overly conservative. In contrast, the DC-PF-based analysis developed here applies to arbitrary ML models, does not depend on the pretrained model, and provides insight into which contingencies may require more iterations or a larger training set.

II. PROBLEM STATEMENT AND FORMULATION

This section defines the AC contingency analysis problem and proposes solving it with a single ML model trained on the basecase topology. A power system can be represented by a graph $\mathcal{G}_0 = (\mathcal{N}, \mathcal{L}_0)$. The set of nodes $\mathcal{N}$ collects the $N+1$ buses indexed by $n = 0, 1, \dots, N$. The set of edges $\mathcal{L}_0 := \{(m, n) : m, n \in \mathcal{N}\}$ collects L transmission lines energized under the basecase system topology. The lines are indexed by $\ell = 1, \dots, L$. We partition $\mathcal{N}$ into three mutually exclusive sets: the singleton of the reference bus indexed by $n = 0$; the set $\mathcal{V}$ of generator buses; and the set of load buses $\mathcal{P}$. Zero-injection buses belong to set $\mathcal{P}$. Buses hosting both generation and loads belong to set $\mathcal{V}$. For each bus $n \in \mathcal{N}$, let

$v_n = V_n \angle \theta_n$ denote the voltage phasor, i_n the injection current phasor, and $s_n = p_n + jq_n$ the complex power injection into bus n . Let vectors $\mathbf{v}$, $\mathbf{i}$, and $\mathbf{s}$ stack the bus voltages, currents, and complex power injections for all $n \in \mathcal{N}$, respectively.

Under the basecase topology, the AC-PF equations can be compactly expressed as

$$\mathbf{s} = \mathbf{v} \odot \bar{\mathbf{i}} = \mathbf{v} \odot (\bar{\mathbf{Y}}_0 \bar{\mathbf{v}}), \quad (1)$$

where $\mathbf{Y}_0$ is the bus admittance matrix, $\odot$ denotes entrywise multiplication, and barred symbols denote complex conjugation. Matrix $\mathbf{Y}_0$ can be expressed as

$$\mathbf{Y}_0 = \sum_{\ell \in \mathcal{L}_0} (y_\ell \mathbf{a}_\ell \mathbf{a}_\ell^\top + y_\ell^s (\mathbf{e}_m \mathbf{e}_m^\top + \mathbf{e}_n \mathbf{e}_n^\top)), \quad (2)$$

where y_ℓ and y_ℓ^s are the series admittance and half shunt admittance of line ℓ , respectively. If line $\ell = (m, n)$ runs between buses m and n , we define vector

$$\mathbf{a}_\ell := \mathbf{e}_m - \mathbf{e}_n \quad \forall \ell \in \mathcal{L}_0, \quad (3)$$

where $\mathbf{e}_m$ is the m -th column of the identity matrix $\mathbf{I}_{N+1}$.

The AC-PF problem is to find a power system state vector $\mathbf{v}$ that satisfies (1), given two specifications per bus. The specification types vary with the bus type: *i*) For the reference bus, we specify V_0 and set $\theta_0 = 0$; *ii*) For PV buses, we specify (p_n, V_n) ; and *iii*) For PQ buses, we specify (p_n, q_n) . Given these $(2N+1)$ real-valued specifications, we can find an AC-PF solution by solving (1) using the Newton-Raphson (NR) or other numerical methods.

Assuming a particular AC-PF solver, define the mapping $G_0 : \mathbb{R}^{2N+1} \rightarrow \mathbb{C}^{N+1}$ from the vector of AC-PF specifications $\mathbf{c} \in \mathbb{R}^{2N+1}$ to the associated AC-PF solution

$$\mathbf{v}_0 := G_0(\mathbf{c}). \quad (4)$$

The subscript 0 emphasizes that this mapping refers to the basecase topology $\mathcal{L}_0$.

In addition to basecase, a system operator is interested in finding the power system state under any single-line contingency. Let ℓ index the outage of line $\ell \in \mathcal{L}_0$. The system topology under contingency ℓ is defined over the edge set

$$\mathcal{L}_\ell := \mathcal{L}_0 \setminus \{\ell\}.$$

After contingency ℓ , the power system graph becomes $\mathcal{G}_\ell = (\mathcal{N}, \mathcal{L}_\ell)$. If graph $\mathcal{G}_\ell$ is disconnected, we call line ℓ a *critical line*. We henceforth assume that contingency analysis does not study the outage of critical lines. To simplify the exposition, let us assume that all lines are non-critical.

As with the basecase topology, the operator can use an AC-PF solver to find the system state $\mathbf{v}_\ell$ under contingency ℓ . Similar to G_0 , the related mapping $G_\ell : \mathbb{R}^{2N+1} \rightarrow \mathbb{C}^{N+1}$ from AC-PF specifications to the post-contingency state is

$$\mathbf{v}_\ell = G_\ell(\mathbf{c}). \quad (5)$$

Clearly, running contingency analysis under the AC-PF model entails solving $(L+1)$ AC-PF problems. Under preventive contingency analysis, all contingencies share the same AC-PF specification vector $\mathbf{c}$. Under corrective contingency analysis, the AC-PF specifications may vary per contingency. Our proposed methodology applies to both settings.

To expedite AC contingency analysis, we first relate each G_ℓ to G_0 . The bus admittance matrix $\mathbf{Y}_\ell$ associated with contingency topology $\mathcal{L}_\ell$ can be expressed as

$$\mathbf{Y}_\ell = \mathbf{Y}_0 - y_\ell \mathbf{a}_\ell \mathbf{a}_\ell^\top - \bar{y}_\ell^s (\mathbf{e}_m \mathbf{e}_m^\top + \mathbf{e}_n \mathbf{e}_n^\top). \quad (6)$$

The contingency state satisfies the modified AC-PF equations

$$\mathbf{s} = \mathbf{v}_\ell \odot (\bar{\mathbf{Y}}_\ell \bar{\mathbf{v}}_\ell). \quad (7)$$

Substituting (6) into (7) and rearranging yields

$$\mathbf{s} + \check{\mathbf{g}}_\ell(\mathbf{v}_\ell) = \mathbf{v}_\ell \odot (\bar{\mathbf{Y}}_0 \bar{\mathbf{v}}_\ell), \quad (8)$$

where the mapping $\check{\mathbf{g}}_\ell(\mathbf{v}_\ell) : \mathbb{C}^{N+1} \rightarrow \mathbb{C}^{N+1}$ is defined as

$$\check{\mathbf{g}}_\ell(\mathbf{v}_\ell) := \mathbf{v}_\ell \odot (\bar{y}_\ell \mathbf{a}_\ell \mathbf{a}_\ell^\top \bar{\mathbf{v}}_\ell + \bar{y}_\ell^s (\mathbf{e}_m \mathbf{e}_m^\top + \mathbf{e}_n \mathbf{e}_n^\top) \bar{\mathbf{v}}_\ell). \quad (9)$$

Vector $\check{\mathbf{g}}_\ell(\mathbf{v}_\ell)$ has only two non-zero entries, corresponding to the terminal buses (m, n) of the outaged line ℓ .

Interestingly, the contingent PF equations in (8) can be interpreted as the basecase PF equations in (1) if complex power injections are modified from $\mathbf{s}$ to $\mathbf{s} + \check{\mathbf{g}}_\ell(\mathbf{v}_\ell)$. A similar concept, oftentimes termed the *compensation trick*, appears in contingency analysis under the linearized DC-PF model [2, Sec. 7.4]. Under the DC-PF model, the compensation trick applies to a system of linear equations. In contrast, the compensation trick here applies to a system of quadratic equations.

Thanks to this interpretation, the system state under contingency ℓ satisfies

$$\mathbf{v}_\ell = G_0 (\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell)), \quad (10)$$

where the mapping $\mathbf{g}_\ell(\mathbf{v}_\ell) : \mathbb{C}^{N+1} \rightarrow \mathbb{R}^{2N+1}$ is defined based on $\check{\mathbf{g}}_\ell(\mathbf{v}_\ell)$ as follows: *i)* For the entries of $\mathbf{c}$ corresponding to active power injections into PQ/PV buses, the mapping $\mathbf{g}_\ell(\mathbf{v}_\ell)$ extracts the real part of $\check{\mathbf{g}}_\ell(\mathbf{v}_\ell)$; *ii)* For the entries of $\mathbf{c}$ corresponding to reactive power injections into PV buses, the mapping $\mathbf{g}_\ell(\mathbf{v}_\ell)$ extracts the imaginary part of $\check{\mathbf{g}}_\ell(\mathbf{v}_\ell)$; and *iii)* For the entries of $\mathbf{c}$ corresponding to voltage magnitudes at the reference and the PV buses, the mapping $\mathbf{g}_\ell(\mathbf{v}_\ell)$ takes zero values. Therefore, depending on whether the terminal buses of line ℓ are PQ, PV, or the reference bus, the vector $\mathbf{g}_\ell(\mathbf{v}_\ell)$ has 2–4 nonzero entries.

Equation (10) establishes that any post-contingency state can be defined with the help of the basecase AC-PF mapping. Unfortunately, finding $\mathbf{v}_\ell$ from (10) is not straightforward, as the state appears on either side of this equation. Nonetheless, one may attempt finding $\mathbf{v}_\ell$ using the fixed-point iteration

$$\mathbf{v}_\ell^{t+1} = G_0 (\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t)). \quad (11)$$

Each iteration t involves two steps: In the first step, we modify 2–4 entries of $\mathbf{c}$ by evaluating $\mathbf{g}_\ell$ at the previous state estimate $\mathbf{v}_\ell^t$. In the second step, we feed the modified specifications to the basecase AC-PF solver and receive the updated state $\mathbf{v}_\ell^{t+1}$. If this iteration converges, its limit point satisfies (10). However, finding $\mathbf{v}_\ell$ using (11) rather than (5) bears no practical advantage. The fixed-point iteration invokes the AC-PF solver G_0 multiple times, which is clearly computationally more expensive than invoking G_ℓ just once. The significance of (11) lies in the insight it provides for translating this fixed-point iteration idea to an ML setting, as described next.

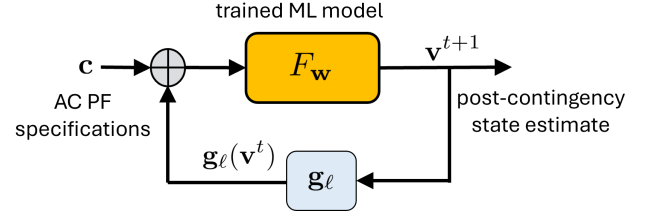


Fig. 1. Fixed-point formulation for predicting a post-contingency state using an ML model trained solely on the basecase topology. The feedback loop depends on the contingency. For a single-line contingency, the mapping $\mathbf{g}_\ell(\mathbf{v}^t)$ is nonzero only at the terminal buses of the outaged line.

To accelerate AC-PF studies, a system operator may use an ML model (such as a DNN, GNN, or Gaussian process) to learn the basecase AC-PF mapping. Suppose we have already trained an ML model to predict AC-PF solutions given a specification vector $\mathbf{c}$ as input. If vector $\mathbf{w}$ collects the ML tunable parameters, we denote the ML mapping as

$$\mathbf{v} = F_{\mathbf{w}}(\mathbf{c}). \quad (12)$$

How about the contingency topologies? Given the large number of transmission lines, training a different ML model for each contingency topology is impractical. The fixed-point iteration in (11) suggests a possibly viable alternative. It suffices to train a *single* ML model under the basecase topology to predict the state under any contingency ℓ by iteratively invoking the basecase ML model as

$$\mathbf{v}_\ell^{t+1} = F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t)). \quad (13)$$

The proposed approach incorporates the basecase ML model into a fixed-point iteration, as shown in Fig. 1.

During training, the proposed approach is meaningful because it avoids training L additional ML models. During inference, the proposed approach can be computationally attractive if the fixed-point iterations converge quickly and the ML inference step is faster than an AC-PF solver. Before evaluating the proposed methodology numerically in Sec. VI, we address the following three questions:

- q1) Does the fixed-point iteration in (13) converge for all contingencies? If so, what is its convergence rate? This question is addressed in Sec. III.
- q2) How can we ensure that the ML model has been trained to predict states accurately when given $\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t)$ as its input for all contingencies ℓ , specification vectors $\mathbf{c}$, and iterations t ? This question is addressed in Sec. IV.
- q3) How does the prediction error of the ML model propagate throughout iterations into the final post-contingency state estimate? This question is addressed in Sec. V.

III. CONVERGENCE RATE ANALYSIS

Establishing the convergence of (13) is challenging because this fixed-point iteration involves an ML model coupled with the nonlinear mapping $\mathbf{g}_\ell(\mathbf{v})$. A natural approach to ensure (13) reaches an equilibrium is to show that $F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}))$ is a contraction mapping with respect to $\mathbf{v}$, upon invoking Banach's fixed-point theorem [27]. To do so, we need to characterize the smoothness of the ML model mapping through

its Lipschitz constant. In [26], we bounded this constant by solving an SDP. Nevertheless, this SDP formulation applies only to DNNs, may be computationally expensive, and may yield overly conservative upper bounds.

Alternatively, if the ML model has learned the AC-PF mapping accurately enough, one may try establishing the convergence of (11) rather than (13). However, proving $G_0(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell))$ is contracting is also technically challenging. Considering the AC-PF solver alone, existing convergence results rely on contractions, energy functions, or monotone operators, but apply only to lossless networks, radial networks, or limited loading conditions [28], [29], [30].

Given this predicament, we approximate the AC-PF mapping in (11) by the linearized mappings corresponding to the DC-PF model. For the basecase, the DC-PF mappings are

$$\mathbf{p} = \mathbf{B}_0 \boldsymbol{\theta}, \quad (14a)$$

$$\mathbf{q} = \mathbf{B}_0 \mathbf{u}, \quad (14b)$$

where vectors $(\mathbf{p}, \mathbf{q})$ collect the active and reactive power injections at all buses, vector $\boldsymbol{\theta}$ collects the bus voltage angles, vector $\mathbf{u}$ collects the bus voltage magnitude deviations from one per unit, and the bus susceptance matrix is defined as

$$\mathbf{B}_0 := \sum_{k \in \mathcal{L}_0} \frac{1}{x_k} \mathbf{a}_k \mathbf{a}_k^\top, \quad (15)$$

where $x_k > 0$ is the reactance of line k . To simplify the exposition and without loss of generality, we use the same matrix $\mathbf{B}_0$ for the two components of the DC-PF model.

Despite its approximate nature, the convergence analysis relying on the DC-PF model reveals the key factors governing the convergence of (11) and (13). Recall that (11) maps AC-PF specifications to complex voltages. Under the DC-PF approximation, the exact AC-PF mapping G_0 decouples into the P - θ and the Q - V mappings in (14). We next study the convergence of these two mappings.

A. Convergence of the P - θ Mapping

If we drop the first row and column from $\mathbf{B}_0$, we obtain the reduced bus susceptance matrix defined as

$$\tilde{\mathbf{B}}_0 := \sum_{k \in \mathcal{L}_0} \frac{1}{x_k} \tilde{\mathbf{a}}_k \tilde{\mathbf{a}}_k^\top, \quad (16)$$

where vector $\tilde{\mathbf{a}}_k$ is obtained from $\mathbf{a}_k$ upon dropping its first entry corresponding to the reference bus. Similar to the AC case, the matrix appearing in the DC-PF model under contingency ℓ is a rank-one modification of $\tilde{\mathbf{B}}_0$. Hence, the reduced P - θ model under contingency ℓ reads as

$$\tilde{\mathbf{p}} = \left(\tilde{\mathbf{B}}_0 - \frac{1}{x_\ell} \tilde{\mathbf{a}}_\ell \tilde{\mathbf{a}}_\ell^\top \right) \tilde{\boldsymbol{\theta}}_\ell, \quad (17)$$

where $\tilde{\mathbf{p}}$ and $\tilde{\boldsymbol{\theta}}$ are N -length vectors excluding the reference bus. Rearranging terms yields

$$\tilde{\boldsymbol{\theta}}_\ell = \tilde{\mathbf{G}}_\ell \tilde{\boldsymbol{\theta}}_\ell + \tilde{\mathbf{B}}_0^{-1} \tilde{\mathbf{p}}, \quad (18)$$

where $\tilde{\mathbf{G}}_\ell := \frac{1}{x_\ell} \tilde{\mathbf{B}}_0^{-1} \tilde{\mathbf{a}}_\ell \tilde{\mathbf{a}}_\ell^\top$. Equation (18) naturally gives rise to the fixed-point iteration

$$\tilde{\boldsymbol{\theta}}_\ell^{t+1} = \tilde{\mathbf{G}}_\ell \tilde{\boldsymbol{\theta}}_\ell^t + \tilde{\mathbf{B}}_0^{-1} \tilde{\mathbf{p}}, \quad (19)$$

which is a discrete-time affine system. Before establishing its stability, we review power transfer distribution factors (PTDFs) and derive a key property instrumental to subsequent analysis.

According to the DC-PF model, the vector $\mathbf{f} \in \mathbb{R}^L$ of active power line flows depends linearly on active power injections

$$\mathbf{f} = \mathbf{S} \mathbf{p}, \quad (20)$$

where $\mathbf{S} := \mathbf{X}^{-1} \tilde{\mathbf{A}} \tilde{\mathbf{B}}_0^{-1}$, the diagonal matrix $\mathbf{X}$ carries the line reactances on its main diagonal, and $\tilde{\mathbf{A}}$ has vectors $\{\tilde{\mathbf{a}}_k\}_{k=1}^L$ as its rows. The (ℓ, m) -th entry of $\mathbf{S}$, denoted by $S_{\ell, m}$, quantifies the sensitivity of line flow ℓ to bus injection m . If one unit of power is transferred from bus i to bus j , the resulting change in line flow ℓ is given by the *power transfer distribution factor*

$$\text{PTDF}_{\ell, i, j} := S_{\ell, i} - S_{\ell, j}. \quad (21)$$

If bus i or j is the reference bus, set $S_{\ell, i} = 0$ or $S_{\ell, j} = 0$, accordingly. We next show a key property of PTDFs.

Theorem 1. *For any non-critical line ℓ , it holds that*

$$|\text{PTDF}_{\ell, i, j}| < 1, \quad \forall i, j \in \mathcal{N}.$$

Although this property is widely used, we could not find a formal proof in the literature. For completeness, a proof of Theorem 1 is provided in the appendix. We next use Th. 1 to establish the convergence of (19). While Th. 1 applies to any PTDF triplet (ℓ, i, j) , the next result requires the property only for the triplet (ℓ, m, n) , where m and n are the terminal buses of line ℓ . We will refer to $\text{PTDF}_{\ell, m, n}$ as the *self-PTDF* of line $\ell = (m, n)$, and denote it with the more compact notation

$$\beta_\ell := \text{PTDF}_{\ell, m, n}. \quad (22)$$

Theorem 2. *The P - θ fixed-point iteration in (19) converges to the solution of (18) for any non-critical line $\ell = (m, n) \in \mathcal{L}$. More specifically, for every $t \geq 1$, it holds that*

$$\|\boldsymbol{\theta}_\ell^t - \boldsymbol{\theta}_\ell\|_2 \leq \beta_\ell^{t-1} \cdot \|\tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell)\|_2, \quad (23)$$

provided that the initial point $\boldsymbol{\theta}_\ell^0$ satisfies $\tilde{\mathbf{a}}_\ell^\top(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell) \neq 0$. The self-PTDF satisfies $\beta_\ell \in (0, 1)$.

Theorem 2 establishes that the fixed-point iteration in (19) converges geometrically for all non-critical lines. It also shows that contingencies on lines with self-PTDF values near one may require more iterations to converge. The condition on the initial point is mild and is satisfied, for example, by $\boldsymbol{\theta}_\ell^0 = \mathbf{0}$. To complete the approximate convergence analysis and address question *q1*), we establish analogous convergence results for the Q - V DC-PF fixed-point iteration. Interested readers can find this analysis in the next subsection; otherwise, we proceed to question *q2*) in Sec. IV.

B. Convergence of the Q - V Mapping

Despite the resemblance between (14a) and (14b), studying the convergence of the Q - V DC-PF fixed-point iteration is slightly more involved. Partition (14b) into PQ/PV buses as

$$\begin{bmatrix} \mathbf{q}_P \\ \mathbf{q}_V \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{PP} & \mathbf{B}_{PV} \\ \mathbf{B}_{VP} & \mathbf{B}_{VV} \end{bmatrix} \begin{bmatrix} \mathbf{u}_P \\ \mathbf{u}_V \end{bmatrix}. \quad (24)$$

For this analysis only, we treat the reference bus as a PV bus. This is because its voltage magnitude is specified and active power injections are immaterial in the $Q-V$ model.

Different from the $P-\theta$ model, here we are given $(\mathbf{q}_P, \mathbf{u}_V)$ and want to find $(\mathbf{q}_V, \mathbf{u}_P)$. In fact, the subvector $\mathbf{u}_P$ is the key unknown in (24). Once we know $\mathbf{u}_P$, we can find $\mathbf{q}_V$ as $\mathbf{q}_V = \mathbf{B}_{VP}\mathbf{u}_P + \mathbf{B}_{VV}\mathbf{u}_V$. Hence, we focus on the top row partition of (24) and rearrange it as

$$\mathbf{B}_{PP}\mathbf{u}_P = \mathbf{q}_P - \mathbf{B}_{PV}\mathbf{u}_V. \quad (25)$$

Equation (25) applies for the basecase. Under contingency ℓ , the non-reduced bus susceptance matrix is modified as

$$\mathbf{B}_\ell = \mathbf{B}_0 - \frac{1}{x_\ell} \mathbf{a}_\ell \mathbf{a}_\ell^\top. \quad (26)$$

Let us partition $\mathbf{a}_\ell$ conformably with (24) as

$$\mathbf{a}_\ell = \begin{bmatrix} \mathbf{a}_{\ell P} \\ \mathbf{a}_{\ell V} \end{bmatrix}.$$

Per (25), the $Q-V$ model under contingency ℓ reads as

$$\left(\mathbf{B}_{PP} - \frac{1}{x_\ell} \mathbf{a}_{\ell P} \mathbf{a}_{\ell P}^\top \right) \mathbf{u}_P = \mathbf{q}_P - \underbrace{\left(\mathbf{B}_{PV} - \frac{1}{x_\ell} \mathbf{a}_{\ell P} \mathbf{a}_{\ell V}^\top \right) \mathbf{u}_V}_{:= \tilde{\mathbf{q}}_\ell^P}. \quad (27)$$

The vector on the right-hand side can be computed beforehand because both $\mathbf{q}_P$ and $\mathbf{u}_V$ are known. We can now convert (27) into a fixed-point iteration to find the unknown voltage magnitudes at PQ buses under contingency ℓ as

$$\mathbf{u}_P^{t+1} = \mathbf{G}_\ell \mathbf{u}_P^t + \mathbf{B}_{PP}^{-1} \tilde{\mathbf{q}}_\ell^P \quad (28)$$

where $\mathbf{G}_\ell := \frac{1}{x_\ell} \mathbf{B}_{PP}^{-1} \mathbf{a}_{\ell P} \mathbf{a}_{\ell P}^\top$. The matrix $\mathbf{B}_{PP}$ is invertible because it is a principal submatrix of $\tilde{\mathbf{B}}_0$. Matrix $\mathbf{G}_\ell$ has rank at most one. Its strictly positive eigenvalue equals to its trace

$$\lambda_\ell = \frac{1}{x_\ell} \mathbf{a}_{\ell P}^\top \mathbf{B}_{PP}^{-1} \mathbf{a}_{\ell P}. \quad (29)$$

We next establish that (28) converges for all non-critical lines.

Theorem 3. *The $Q-V$ fixed-point iteration in (28) converges to the solution of (27) for any non-critical line $\ell = (m, n) \in \mathcal{L}_0$. In particular, the eigenvalue of matrix $\mathbf{G}_\ell$ in (28) satisfies $0 \leq \lambda_\ell < 1$. Consequently, for every $t \geq 1$,*

$$\|\mathbf{u}_{\ell, P}^t - \mathbf{u}_{\ell, P}\|_2 \leq \lambda_\ell^{t-1} \|\mathbf{G}_\ell(\mathbf{u}_{\ell, P}^0 - \mathbf{u}_{\ell, P})\|_2. \quad (30)$$

We finish this section by illustrating the distribution of the contraction constants β_ℓ involved in the $P-\theta$ model (Th. 2) and λ_ℓ in the $Q-V$ model (Th. 3) for the Texas7k benchmark system. This is a synthesized 6,717-bus system with 8,079 non-critical lines out of 9,140 transmission lines. Figure 2 shows the distributions of these spectral radii across the non-critical lines, confirming they are all less than one.

IV. ADMISSIBLE REGION FOR ML MODEL INPUTS

Suppose an operator wants to study a power system for all specification vectors $\mathbf{c}$ in a given set $\mathcal{C} \subset \mathbb{R}^{2N+1}$. If the operator is interested only in the basecase topology, they can train an ML model to predict basecase AC-PF solutions for sufficiently many specification scenarios drawn from $\mathcal{C}$.

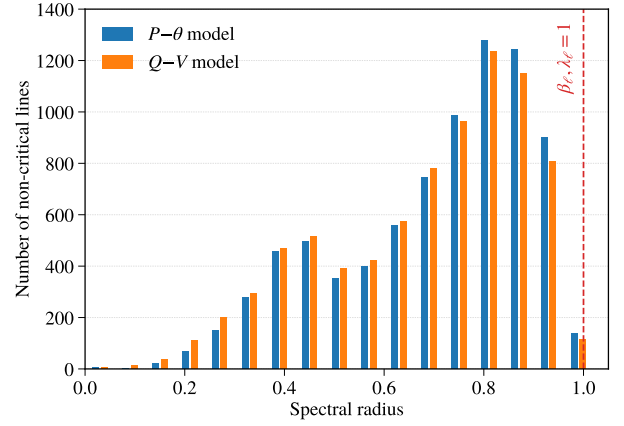


Fig. 2. Distribution of contraction constants involved in the $P-\theta$ and the $Q-V$ models across all 8,079 non-critical lines of the Texas7k system.

In this case, the *training region* of the ML model coincides with the *specification region* of the AC-PF study. However, if the operator intends to use the same model as in (13) to conduct contingency analysis over the specification region $\mathcal{C}$, the model must be trained over an expanded training region $\mathcal{C}'$, with $\mathcal{C} \in \mathcal{C}'$. This section explains how to find $\mathcal{C}'$ given $\mathcal{C}$.

Why should the ML model be trained on an expanded training set? For the fixed-point iteration of (13) to succeed, the ML model should accurately predict the AC-PF solutions for all modified specification vectors

$$\mathbf{c}_\ell^t := \mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t). \quad (31)$$

This should hold for all $\mathbf{c} \in \mathcal{C}$, all contingencies $\ell \in \mathcal{L}$, and across iterations indexed by t . Therefore, the training set of the ML model $\mathcal{C}'$ should include all possible values of $\mathbf{c}_\ell^t$ from (31). If we set $\mathbf{v}_\ell^t = \mathbf{1}$ (flat voltage profile) in (31), we get $\mathbf{c}_\ell^t = \mathbf{c}$. This implies that if $\mathbf{c} \in \mathcal{C}$, then $\mathbf{c} \in \mathcal{C}'$, and thus, $\mathcal{C} \subset \mathcal{C}'$. The cardinal question is how to find $\mathcal{C}'$.

Determining $\mathcal{C}'$ to include all vectors $\mathbf{c}_\ell^t$ is a formidable task because (13) is nonlinear, iterative, and involves the ML model. To keep the analysis tractable, we again resort to the DC-PF models of Sec. III. Moreover, we postulate that the specification set admits the Cartesian-product structure

$$\mathcal{C} = \mathcal{C}_p \times \mathcal{C}_q \times \mathcal{C}_u, \quad (32)$$

where $\mathcal{C}_p$, $\mathcal{C}_q$, and $\mathcal{C}_u$ are the sets of active/reactive power and voltage magnitude specifications, respectively.

Due to the decoupled nature of the DC-PF models, the training set $\mathcal{C}'$ inherits the same Cartesian-product structure

$$\mathcal{C}' = \mathcal{C}'_p \times \mathcal{C}'_q \times \mathcal{C}'_u. \quad (33)$$

Because $\mathbf{g}_\ell(\mathbf{v})$ modifies only power injections and no voltage magnitudes, we can safely set $\mathcal{C}'_u = \mathcal{C}_u$. In the subsequent analysis, we rely on the $P-\theta$ model to approximate $\mathcal{C}'_p$ from $\mathcal{C}_p$, and on the $Q-V$ model to approximate $\mathcal{C}'_q$ from $\mathcal{C}_q$.

Using the $P-\theta$ model, we first derive a sufficient condition for $\mathcal{C}'_p$ to be a valid training set.

Lemma 1. *According to the $P-\theta$ model, if the training set $\mathcal{C}'_p$ includes the points*

$$\mu \tilde{\mathbf{p}} + (1 - \mu) \left(\mathbf{I}_N + \frac{1}{1 - \beta_\ell} \tilde{\mathbf{G}}_\ell^\top \right) \tilde{\mathbf{p}}$$

for all $\mu \in [0, 1]$, $\tilde{\mathbf{p}} \in \mathcal{C}_p$, and contingencies ℓ , then $\mathcal{C}_p \subset \mathcal{C}'_p$.

Based on Lemma 1, we next provide an outer approximation of $\mathcal{C}'_p$ for the case that $\mathcal{C}_p$ takes a polytopic form as

$$\mathcal{C}_p := \{\tilde{\mathbf{p}} : \mathbf{C}_p \tilde{\mathbf{p}} \leq \mathbf{d}_p\}. \quad (34)$$

where $\mathbf{C}_p \in \mathbb{R}^{I_p \times N}$. This polytopic form is reasonable, since $\mathcal{C}_p$ can be specified by imposing limits on active injections and line flows, according to (20).

According to Lemma 1, the set $\mathcal{C}'_p$ is the union of L polytopes. Because computing $\mathcal{C}'_p$ exactly is difficult, we resort to an outer convex approximation $\hat{\mathcal{C}}_p$ of $\mathcal{C}'_p$, so that $\mathcal{C}_p \subset \mathcal{C}'_p \subseteq \hat{\mathcal{C}}_p$. Aiming for a computationally tractable solution, we assume $\hat{\mathcal{C}}_p$ is also a polytope described by the same matrix $\mathbf{C}_p$ as

$$\hat{\mathcal{C}}_p := \{\tilde{\mathbf{p}} : \mathbf{C}_p \tilde{\mathbf{p}} \leq \hat{\mathbf{d}}_p\}. \quad (35)$$

The goal is to determine $\hat{\mathbf{d}}_p$ so that $\mathcal{C}'_p \subseteq \hat{\mathcal{C}}_p$. A sufficient condition for this to happen is that

- c1) $\tilde{\mathbf{p}} \in \hat{\mathcal{C}}_p$ for all $\tilde{\mathbf{p}} \in \mathcal{C}_p$; and
- c2) $(\mathbf{I}_N + \frac{1}{1-\beta_\ell} \tilde{\mathbf{G}}_\ell)^\top \tilde{\mathbf{p}} \in \hat{\mathcal{C}}_p$ for all $\tilde{\mathbf{p}} \in \mathcal{C}_p$ and for all ℓ .

For condition c1) to be met, we must ensure that $\hat{\mathbf{d}}_p \geq \mathbf{d}_p$. Regarding c2), we first compute the following value for each constraint $i \in \{1, \dots, I_p\}$ defining the polytope $\mathcal{C}_p$:

$$\gamma_{p,i} = \max_{\ell} \max_{\tilde{\mathbf{p}} \in \mathcal{C}_p} \mathbf{e}_i^\top \mathbf{C}_p \left(\mathbf{I}_N + \frac{1}{1-\beta_\ell} \tilde{\mathbf{G}}_\ell \right)^\top \tilde{\mathbf{p}}. \quad (36)$$

To meet c2), we must select the i -th entry of $\hat{\mathbf{p}}_p$ as $\hat{d}_{p,i} \geq \gamma_{p,i}$. To meet both c1) and c2), we can set

$$\hat{\mathbf{d}}_p = \max\{d_{p,i}, \gamma_{p,i}\},$$

where $\tilde{d}_{p,i}$ is the i -th entry of $\mathbf{d}_p$. Therefore, computing $\hat{\mathbf{d}}_p$ entails solving the $I_p \times L$ linear programs (LPs) in (36). The set $\hat{\mathcal{C}}_p$ is an outer approximation of $\mathcal{C}'_p$ because the latter may not necessarily include the entire line segment between $(\mathbf{I}_N + \frac{1}{1-\beta_\ell} \tilde{\mathbf{G}}_\ell)^\top \tilde{\mathbf{p}}$ and $(\mathbf{I}_N + \frac{1}{1-\beta_{\ell'}} \tilde{\mathbf{G}}_{\ell'})^\top \tilde{\mathbf{p}}$ for all line pairs (ℓ, ℓ') .

A similar analysis can be carried out for reactive power injections to obtain $\mathcal{C}'_q$ from $\mathcal{C}_q$ to obtain the ensuing result.

Lemma 2. According to the $Q-V$ model, if the training set $\mathcal{C}'_q$ includes the points

$$\mu \mathbf{q}_P + (1-\mu) \left(\left(\mathbf{I} + \frac{\mathbf{G}_\ell^\top}{1-\lambda_\ell} \right) \mathbf{q}_P + \left(\frac{\mathbf{a}_\ell \mathbf{p} \mathbf{a}_\ell^\top}{x_\ell} - \mathbf{G}_\ell^\top \mathbf{B}_{PV} \right) \mathbf{u}_V \right)$$

for all $\mu \in [0, 1]$, $\mathbf{q}_P \in \mathcal{C}_q$, and ℓ , then $\mathcal{C}_q \subset \mathcal{C}'_q$.

Similar to the $P-\theta$ model, we can use Lemma 2 to derive an outer polytopic approximation $\hat{\mathcal{C}}_q$ of $\mathcal{C}'_q$ if the original specification set $\mathcal{C}_q$ admits the polytopic form

$$\mathcal{C}_q := \{\mathbf{q}_P : \mathbf{C}_q \mathbf{q}_P \leq \mathbf{d}_q\}. \quad (37)$$

More specifically, we postulate that

$$\hat{\mathcal{C}}_q := \{\mathbf{q}_P : \mathbf{C}_q \mathbf{q}_P \leq \hat{\mathbf{d}}_q\}, \quad (38)$$

where $\mathbf{C}_q \in \mathbb{R}^{I_q \times N}$. We then compute the following value for each constraint $i \in \{1, \dots, I_q\}$ defining $\mathcal{C}_q$:

$$\gamma_{q,i} := \max_{\ell} \max_{\substack{\mathbf{q}_P \in \mathcal{C}_q \\ \mathbf{u}_V \in \mathcal{C}_u}} \mathbf{e}_i^\top \mathbf{C}_q \left(\mathbf{I} + \frac{\mathbf{G}_\ell^\top}{1-\lambda_\ell} \right) \mathbf{q}_P$$

$$+ \mathbf{e}_i^\top \mathbf{C}_q \left(\frac{\mathbf{a}_\ell \mathbf{p} \mathbf{a}_\ell^\top}{x_\ell} - \mathbf{G}_\ell^\top \mathbf{B}_{PV} \right) \mathbf{u}_V$$

Finally, we can set the i -th entry of $\hat{\mathbf{d}}_q$ as

$$\hat{d}_{q,i} = \max\{d_{q,i}, \gamma_{q,i}\}$$

for all $i \in \{1, \dots, I_q\}$. This ensures that $\mathcal{C}'_q \subseteq \hat{\mathcal{C}}_q$, so we can safely use $\hat{\mathcal{C}}_q$ as the expanded set for training the ML model. If $\mathcal{C}_u$ is also a polytope, then finding all $\gamma_{q,i}$'s entails solving $I_q \times L$ LPs. For example, if $\mathcal{C}_u$ stems from lower and upper limits on voltage magnitudes at PV buses, then $\mathcal{C}_u$ takes the form of a hyperrectangle.

In a nutshell, we have provided a practical method to compute the expanded training set $\mathcal{C}'$. If the original sets $\mathcal{C}_p$ and $\mathcal{C}_q$ are polytopes with I_p and I_q constraints, respectively, the operator can determine the sets $\hat{\mathcal{C}}_p$ and $\hat{\mathcal{C}}_q$ by solving $(I_p + I_q)L$ linear programs. Although $\hat{\mathcal{C}}$ is an outer approximation and may be conservative, our numerical tests show that an ML model with only $\mathcal{O}(N^2)$ parameters can still be trained accurately over this expanded region. This provides an approximate answer for question q2). We next focus on q3).

V. PREDICTION ACCURACY OF FIXED-POINT SOLUTION

Thus far, we have characterized the convergence of the fixed-point iteration and identified the specification set over which the ML model should be trained. The analysis has presumed that the ML model $F_{\mathbf{w}}(\mathbf{c})$ approximates well the basecase AC-PF mapping $G_0(\mathbf{c})$, so that the ML-based fixed-point iterates of (13) remain close to the AC-PF-based fixed-point iterates of (11). This section establishes that the two trajectories indeed remain close to each other.

For this analysis, we must quantify the sensitivity of the ML mapping $F_{\mathbf{w}}$ to voltage phasors. This sensitivity is formally captured by the Lipschitz constant L_ℓ as

$$\|F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v})) - F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}'))\|_2 \leq L_\ell \|\mathbf{v} - \mathbf{v}'\|_2 \quad (39)$$

for all $\mathbf{v}$ and $\mathbf{v}'$ in a voltage region $\mathcal{R}$. The region $\mathcal{R}$ is defined so that $\mathbf{c} + \mathbf{g}_\ell(\mathbf{v})$ and $\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}')$ belong to $\mathcal{C}'$ for all $\mathbf{c} \in \mathcal{C}$. The constant L_ℓ relates to the Jacobian of the ML model

$$L_\ell = \sup_{\mathbf{v} \in \mathcal{R}} \|\nabla_{\mathbf{v}} F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}))\|_2. \quad (40)$$

Our analysis requires that the ML mapping be contracting in $\mathbf{v}$ as stated next.

Assumption 1. The ML mapping is a contraction in terms of $\mathbf{v}$, so that the Lipschitz constant in (39) satisfies $L_\ell < 1$ for every $\mathbf{c} \in \mathcal{C}$ and every contingency ℓ .

In [26], we proposed an SDP formulation for verifying that a DNN trained on the basecase topology satisfies Assumption 1. We will further need to quantify the approximation error between the ML model and the AC-PF mapping as formalized in the ensuing assumption.

Assumption 2. Suppose the ML model approximates the basecase AC-PF mapping over the set $\mathcal{C}'$ with accuracy $\epsilon > 0$

$$\|F_{\mathbf{w}}(\mathbf{c}) - G_0(\mathbf{c})\|_2 \leq \epsilon, \quad \forall \mathbf{c} \in \mathcal{C}'. \quad (41)$$

We can now bound the error between the ML-based and the AC-PF-based fixed-point trajectories, reproduced here as

$$\mathbf{v}_\ell^{t+1} = G_0(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t)), \quad (42a)$$

$$\hat{\mathbf{v}}_\ell^{t+1} = F_w(\mathbf{c} + \mathbf{g}_\ell(\hat{\mathbf{v}}_\ell^t)). \quad (42b)$$

Lemma 3. *Under Assumptions 1–2, the Euclidean distance*

$$e_\ell^t := \|\hat{\mathbf{v}}_\ell^t - \mathbf{v}_\ell^t\|_2 \quad (43)$$

between the fixed-point iterates in (42) satisfies

$$e_\ell^t \leq \left(\frac{1 - L_\ell^t}{1 - L_\ell} \right) \epsilon \leq \frac{\epsilon}{1 - L_\ell} \quad (44)$$

if the iterates are initialized at the same point $\mathbf{v}_\ell^0 = \hat{\mathbf{v}}_\ell^0$.

Lemma 3 establishes that although the distance between the two trajectories increases with t , it remains bounded. If the AC-PF fixed-point iteration converges to the true post-contingency state, the ML-based fixed-point iteration converges within a neighborhood of that state. According to (44), the radius of the neighborhood depends on the prediction accuracy ϵ of the ML model and its Lipschitz constant L_ℓ , which varies across contingencies. Because L_ℓ may be conservative, the resulting error bound can overestimate the actual distance between the two solutions. The numerical tests of the next section show that the solution of (13) remains close to the post-contingency state across almost all contingencies.

VI. NUMERICAL TESTS

We evaluated the proposed methodology on the 6,717-bus Texas7k benchmark system. To learn the basecase AC-PF mapping, we trained a fully connected DNN with two hidden layers, each containing 6,717 neurons, with ReLU activation. We trained the DNN using the Adam optimizer with a learning rate of 4×10^{-5} , and terminated training was the loss gradient norm fell below 10^{-4} . We conducted all numerical tests in the same RunPod environment on a compute node equipped with an NVIDIA B200 GPU and an Intel Xeon 6960P CPU.

The Texas7k dataset provides one year of operating scenarios at a 5-min resolution. Using k -medoids clustering, we selected 2,000 representative scenarios. We selected 200 of these to define the specification set $\mathcal{C}$. We constructed the corresponding polytope using voltage, power injection, and line flow limits from these 200 scenarios. Following the procedure described in Sec. IV, we constructed the set $\hat{\mathcal{C}}$ to represent the training region. We generated 120,000 scenarios from $\hat{\mathcal{C}}$ and combined them with the remaining 1,800 basecase scenarios to train the DNN on both basecase operating conditions and the specifications encountered during the iterations of (13).

We first examined the convergence of (13). For every scenario, the iteration was initialized at the basecase DNN solution, and convergence was declared when the successive voltage difference norm fell below 10^{-5} . To accelerate the fixed-point iteration, we applied Anderson acceleration (AA) with memory 3; see [31]. We ordered line contingencies by their iteration count to convergence, averaged across scenarios. For cleaner visualization, we sampled the ranked contingencies in increments of 300 and included the last (slowest) one. Figure 3 shows the Euclidean distance between the predicted

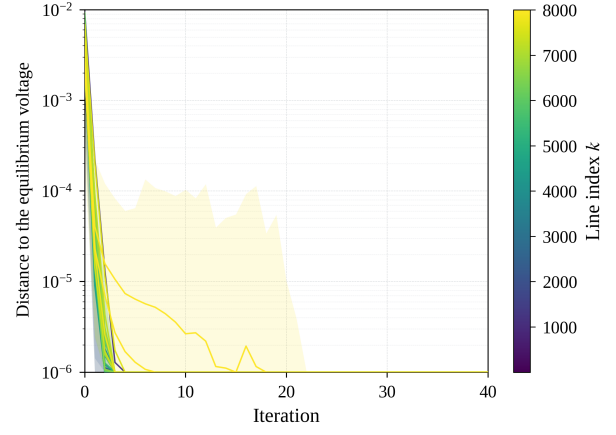


Fig. 3. Convergence of the DNN fixed-point iteration to its equilibrium with Anderson acceleration across different contingencies. The contingencies were ordered according to empirical convergence rate and subsampled every 300.

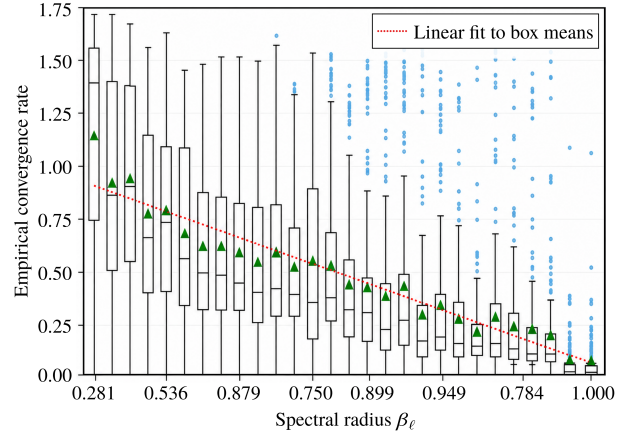


Fig. 4. Distribution of empirical convergence rate over testing scenarios across subsampled contingencies, plotted against the corresponding spectral radii β_ℓ .

and the true post-contingency AC states across iterations. All contingencies converged within 20 iterations.

In the second test, we explored the relationship between the empirical convergence rate and the spectral matrix radius of the P - θ model. We ordered contingencies by β_ℓ and sampled them again in increments of 300. The empirical convergence rate was obtained from the negative slope of the fitted line to the convergence curve in Fig. 3. Figure 4 shows the resulting distribution of empirical rates. In general, the fitted trend decreases as the radius increases, which agrees with Th. 2.

To assess prediction accuracy, the top panel of Fig. 5 shows the MSE between the true and predicted post-contingency states across scenarios for the basecase and 29 contingencies, ordered by β_ℓ and subsampled every 300. The bottom panel shows the NMAE of the predicted AC-PF specifications, obtained by substituting the predicted states into the AC-PF equations. The MSE remains below 0.3% and NMAE below 6% across contingencies, comparable to the basecase.

Finally, we compared the compute performance of the proposed approach with NR and the DC-based solver over all testing scenarios and non-critical contingencies, totaling 1.6M instances. The DNN fixed-point iteration with Anderson acceleration was evaluated on an NVIDIA B200 GPU, while

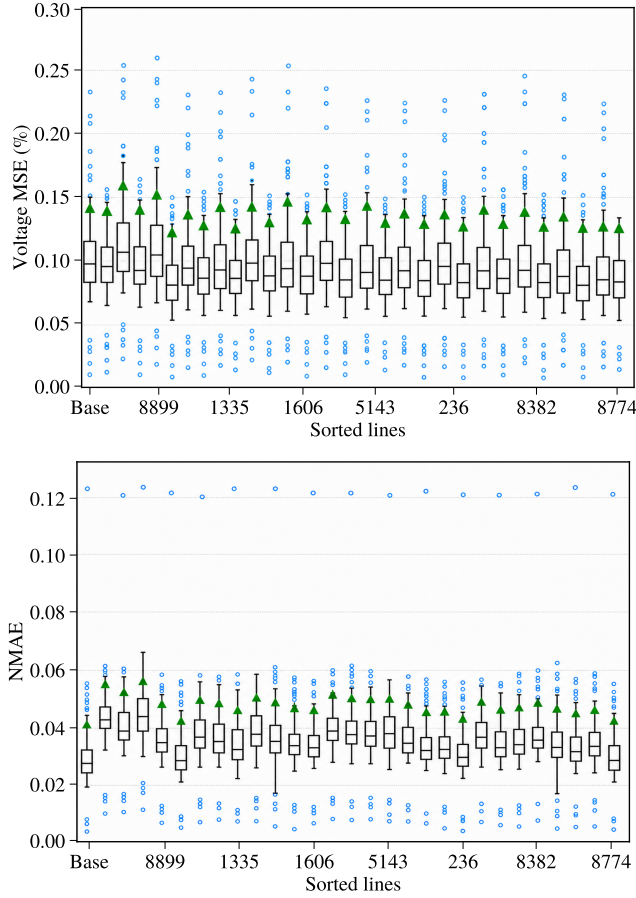


Fig. 5. Box plots of the MSE on predicted voltages (top) and NMAE on predicted AC-PF specifications (bottom) across tested scenarios, for the basecase and 29 contingencies. The method features similar errors across the basecase and the selected contingencies.

NR and the DC-based solver were run on the CPU, with NR contingencies parallelized across 8 workers. The top panel of Fig. 6 compares the per-instance runtime distributions. The proposed method is substantially faster than NR, while the DC-based solver remains fastest. However, this speed comes at the expense of accuracy. The bottom panel of Fig. 6 compares the voltage MSE distributions relative to the NR solution, showing substantially lower errors for the proposed method than for the DC-based solver. The proposed method offers a favorable tradeoff between the speed of the DC model and NR's accuracy.

VII. CONCLUSIONS AND FUTURE WORK

We have put forth a fixed-point formulation for AC contingency analysis using a single ML model trained solely on basecase data. We established convergence guarantees and characterized convergence rates under the DC-PF approximation. We also derived an expanded training region to ensure the DNN accurately predicts AC-PF throughout the fixed-point iterations. We further established bounds on the resulting prediction errors. Numerical tests on the Texas7k system demonstrate convergence across all non-critical line outages, with prediction errors remaining close to the basecase

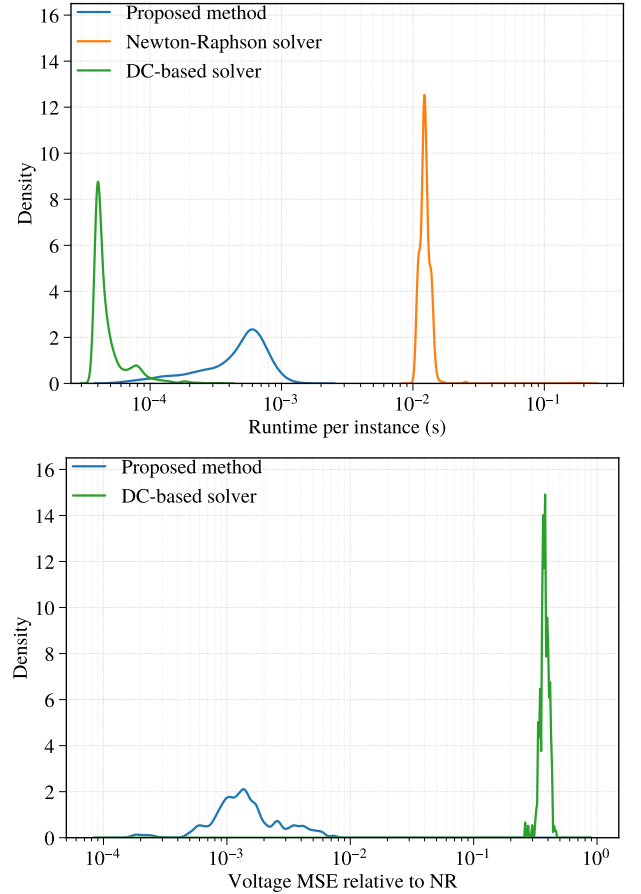


Fig. 6. The proposed method offers a favorable trade-off between runtime (top) and accuracy (bottom) when compared to the DC solver against the NR solution. Histograms are computed across scenarios and Texas7k contingencies on the system.

error. An interesting direction for future work is to extend the formulation to $N - k$ contingencies.

APPENDIX

Proof of Theorem 1: Consider a unit power transfer from the source bus i to the sink bus j , so that $\mathbf{p} = \mathbf{e}_i - \mathbf{e}_j$ in (14a). By definition, the power flow on any line $\ell \in \mathcal{L}_0$ is $\text{PTDF}_{\ell,i,j}$.

We will first show that θ_i is the largest voltage angle and θ_j is the smallest voltage angle across all buses. Recall that the power injection into any bus $k \in \mathcal{N}$ satisfies [cf. (14a)]

$$p_k = \sum_{o:(k,o) \in \mathcal{L}_0} \frac{\theta_k - \theta_o}{x_{ko}}. \quad (45)$$

Because $p_k = 0$ for all $k \neq i, j$, it follows that θ_k cannot be the largest (smallest) angle because otherwise all numerators in the sum of (45) would be positive (negative). Therefore, it is either $\theta_i > \theta_k > \theta_j$ or $\theta_j > \theta_k > \theta_i$ for all $k \neq i, j$. Only the first option can explain $p_i = 1$ and $p_j = -1$ in (45).

Consider now the terminal buses of line $\ell = (m, n)$ and suppose $\theta_m > \theta_n$. Select a voltage angle value $\theta_0 \in (\theta_n, \theta_m)$, and define the set of buses $\mathcal{S} := \{q \in \mathcal{N} : \theta_q \geq \theta_0\}$. By construction, bus m belongs to $\mathcal{S}$, and bus n belongs to its complement set $\bar{\mathcal{S}}$. It also holds that $i \in \mathcal{S}$ and $j \in \bar{\mathcal{S}}$.

Using (45), sum the power injections into the buses in $\mathcal{S}$

$$\sum_{k \in \mathcal{S}} p_k = \sum_{k \in \mathcal{S}} \sum_{o \in \mathcal{S}: (k,o) \in \mathcal{L}_0} \frac{\theta_k - \theta_o}{x_{ko}} + \sum_{k \in \mathcal{S}} \sum_{r \in \bar{\mathcal{S}}: (k,r) \in \mathcal{L}_0} \frac{\theta_k - \theta_r}{x_{kr}}.$$

Each p_k is expressed as a sum of the line flows leaving bus k . Flows are grouped into those terminating at a bus within $\mathcal{S}$, and those terminating at a bus within $\bar{\mathcal{S}}$. The first group of flows internal to $\mathcal{S}$ has zero sum. This is because every flow $(\theta_k - \theta_o)/x_{ko}$ is canceled by its reverse flow $(\theta_o - \theta_k)/x_{ko}$. Hence,

$$\sum_{\substack{(k,r) \in \mathcal{L}_0 \\ k \in \mathcal{S}, r \in \bar{\mathcal{S}}}} \frac{\theta_k - \theta_r}{x_{kr}} = 1.$$

By the definition of $\mathcal{S}$, it holds $\theta_k > \theta_r$ for all $k \in \mathcal{S}$ and $r \in \bar{\mathcal{S}}$, so that all terms in the previous summand are nonnegative. Given they sum up to one, all terms must be smaller than one. The line of interest ℓ belongs to this cut, and so $f_\ell = (\theta_m - \theta_n)/x_{mn} \in (0, 1]$.

Thus far, we assumed that $\theta_m > \theta_n$. If $\theta_m < \theta_n$, then it follows similarly that $i, n \in \mathcal{S}$ and $j, m \in \bar{\mathcal{S}}$. In this case, it follows that $f_\ell \in [-1, 0)$. Combining the two cases proves that $|\text{PTDF}_{\ell,i,j}| \leq 1$. The claim holds even if $i = m$ and $j = n$ for the self-PTDF. If line ℓ is the only line crossing the cut between $\mathcal{S}$ and $\bar{\mathcal{S}}$, then $|\text{PTDF}_{\ell,i,j}| = 1$ and ℓ constitutes a critical line. ■

Proof of Theorem 2: The matrix $\tilde{\mathbf{G}}_\ell$ has rank one. Hence, its unique nonzero eigenvalue equals the trace of $\tilde{\mathbf{G}}_\ell$:

$$\begin{aligned} \tilde{\lambda}_\ell &= \frac{\tilde{\mathbf{a}}_\ell^\top \tilde{\mathbf{B}}_0^{-1} \tilde{\mathbf{a}}_\ell}{x_\ell} = \frac{\tilde{\mathbf{a}}_\ell^\top \tilde{\mathbf{B}}_0^{-1} \mathbf{e}_m}{x_\ell} - \frac{\tilde{\mathbf{a}}_\ell^\top \tilde{\mathbf{B}}_0^{-1} \mathbf{e}_n}{x_\ell} \\ &= S_{\ell,m} - S_{\ell,n} = \text{PTDF}_{\ell,m,n} = \beta_\ell. \end{aligned}$$

The first equality yields $\tilde{\lambda}_\ell > 0$ because $\tilde{\mathbf{B}}_0$ is a positive definite matrix. The second equality follows from the definition of $\tilde{\mathbf{a}}_\ell$ in (3) together with the convention for the reference bus, whereby $S_{\ell,m}$ or $S_{\ell,n}$ is set to zero whenever bus m or n is the reference bus, respectively. The third equality follows from the definition of $\mathbf{S}$ in (20). The fourth equality follows from the PTDF definition in (21). By Theorem 1, we get

$$\tilde{\lambda}_\ell = \text{PTDF}_{\ell,m,n} = \beta_\ell < 1.$$

Subtracting (18) from (19) provides

$$\boldsymbol{\theta}_\ell^{t+1} - \boldsymbol{\theta}_\ell = \tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^t - \boldsymbol{\theta}_\ell).$$

Iterating this recursion yields

$$\boldsymbol{\theta}_\ell^{t+1} - \boldsymbol{\theta}_\ell = \tilde{\mathbf{G}}_\ell^t(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell) = \beta_\ell^t \tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell),$$

where the second equality follows from the rank-one property of $\tilde{\mathbf{G}}_\ell$. Taking Euclidean norms on both sides gives

$$\|\boldsymbol{\theta}_\ell^t - \boldsymbol{\theta}_\ell\|_2 \leq \beta_\ell^{t-1} \|\tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell)\|_2.$$

Hence, the fixed-point iteration in (19) converges to $\boldsymbol{\theta}_\ell$ because $\beta_\ell < 1$, provided that $\tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell) \neq \mathbf{0}$. By the definition of $\tilde{\mathbf{G}}_\ell$ in (18), the condition $\tilde{\mathbf{G}}_\ell(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell) \neq \mathbf{0}$ is equivalent to $\tilde{\mathbf{a}}_\ell^\top(\boldsymbol{\theta}_\ell^0 - \boldsymbol{\theta}_\ell) \neq 0$. ■

Proof of Theorem 3: We study three cases, depending on the types of terminal buses on the contingent line $\ell = (m, n)$.

Case 1: Both terminal buses are PV ($m, n \in \mathcal{V}$). Because $\mathbf{a}_{\ell_P} = \mathbf{0}$, we get $\mathbf{G}_\ell = \mathbf{0}$, $\lambda_\ell = 0$, and hence, the fixed-point iteration in (28) reaches the solution after a single update.

Case 2: Both terminal buses are PQ ($m, n \in \mathcal{P}$). Then

$$\mathbf{a}_{\ell_P} = \mathbf{e}_m - \mathbf{e}_n.$$

To relate the quantity in (29) to the self-PTDF, partition the reduced bus susceptance matrix into PQ and PV buses:

$$\tilde{\mathbf{B}}_0 = \begin{bmatrix} \mathbf{B}_{\mathcal{P}\mathcal{P}} & \tilde{\mathbf{B}}_{\mathcal{P}\mathcal{V}} \\ \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{P}} & \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{V}} \end{bmatrix}.$$

The reference bus has been excluded here. Let its inverse be partitioned conformably as

$$\tilde{\mathbf{C}}_0 := \tilde{\mathbf{B}}_0^{-1} = \begin{bmatrix} \mathbf{C}_{\mathcal{P}\mathcal{P}} & \tilde{\mathbf{C}}_{\mathcal{P}\mathcal{V}} \\ \tilde{\mathbf{C}}_{\mathcal{V}\mathcal{P}} & \tilde{\mathbf{C}}_{\mathcal{V}\mathcal{V}} \end{bmatrix}.$$

The block-inverse formula provides that

$$\mathbf{C}_{\mathcal{P}\mathcal{P}} = \left(\mathbf{B}_{\mathcal{P}\mathcal{P}} - \tilde{\mathbf{B}}_{\mathcal{P}\mathcal{V}} \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{V}}^{-1} \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{P}} \right)^{-1}. \quad (46)$$

Since $\tilde{\mathbf{B}}_0 \succ \mathbf{0}$, its Schur complement satisfies

$$\mathbf{0} \prec \mathbf{B}_{\mathcal{P}\mathcal{P}} - \tilde{\mathbf{B}}_{\mathcal{P}\mathcal{V}} \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{V}}^{-1} \tilde{\mathbf{B}}_{\mathcal{V}\mathcal{P}} \preceq \mathbf{B}_{\mathcal{P}\mathcal{P}}.$$

Inverting the two matrices yields $\mathbf{B}_{\mathcal{P}\mathcal{P}}^{-1} \preceq \mathbf{C}_{\mathcal{P}\mathcal{P}}$, and so $\mathbf{a}_{\ell_P}^\top \mathbf{B}_{\mathcal{P}\mathcal{P}}^{-1} \mathbf{a}_{\ell_P} \leq \mathbf{a}_{\ell_P}^\top \mathbf{C}_{\mathcal{P}\mathcal{P}} \mathbf{a}_{\ell_P}$. Therefore, we get that

$$\begin{aligned} \lambda_\ell &= \frac{1}{x_\ell} \mathbf{a}_{\ell_P}^\top \mathbf{B}_{\mathcal{P}\mathcal{P}}^{-1} \mathbf{a}_{\ell_P} \leq \frac{1}{x_\ell} \mathbf{a}_{\ell_P}^\top \mathbf{C}_{\mathcal{P}\mathcal{P}} \mathbf{a}_{\ell_P} \\ &= \frac{1}{x_\ell} \mathbf{a}_\ell^\top \tilde{\mathbf{C}}_0 \mathbf{a}_\ell = \frac{1}{x_\ell} \tilde{\mathbf{a}}_\ell^\top \tilde{\mathbf{B}}_0^{-1} \tilde{\mathbf{a}}_\ell = \text{PTDF}_{\ell,m,n} < 1. \end{aligned}$$

The second equality holds because all the non-zero entries of $\tilde{\mathbf{a}}_\ell$ are included in the subvector $\mathbf{a}_{\ell_P}$. Because matrix $\mathbf{B}_{\mathcal{P}\mathcal{P}}$ is positive semidefinite, we also get that $\lambda_\ell \geq 0$.

Case 3: One terminal bus is PQ and the other is PV ($m \in \mathcal{P}, n \in \mathcal{V}$). In this case, we get that $\mathbf{a}_{\ell_P} = \pm \mathbf{e}_m$ and

$$\lambda_\ell = \frac{1}{x_\ell} \mathbf{e}_m^\top \mathbf{B}_{\mathcal{P}\mathcal{P}}^{-1} \mathbf{e}_m.$$

The matrix $\mathbf{B}_{\mathcal{P}\mathcal{P}}$ can be decomposed as

$$\mathbf{B}_{\mathcal{P}\mathcal{P}} = \frac{1}{x_\ell} \mathbf{e}_m \mathbf{e}_m^\top + \sum_{k \in \mathcal{L}_0 \setminus \{\ell\}} \frac{1}{x_k} \mathbf{a}_{k_P} \mathbf{a}_{k_P}^\top. \quad (47)$$

Because line ℓ is non-critical, the power system remains connected after its removal. Therefore, the second term on the right-hand side of (47) is a positive definite matrix, and so

$$\frac{1}{x_\ell} (\mathbf{e}_m^\top \mathbf{z})^2 < \mathbf{z}^\top \mathbf{B}_{\mathcal{P}\mathcal{P}} \mathbf{z}$$

for any $\mathbf{z} \neq \mathbf{0}$. According to the variational identity for a positive definite matrix, we have that

$$\mathbf{e}_m^\top \mathbf{B}_{\mathcal{P}\mathcal{P}}^{-1} \mathbf{e}_m = \max_{\mathbf{z} \neq \mathbf{0}} \frac{(\mathbf{e}_m^\top \mathbf{z})^2}{\mathbf{z}^\top \mathbf{B}_{\mathcal{P}\mathcal{P}} \mathbf{z}} < x_\ell,$$

which implies that $\lambda_\ell \in (0, 1)$.

We have established that for all three cases of terminal bus types, the spectral radius of $\mathbf{G}_\ell$ satisfies $0 \leq \lambda_\ell < 1$ for any non-critical line. Finally, subtracting (27) from (28) and using the rank-one identity $\mathbf{G}_\ell^t = \lambda_\ell^{t-1} \mathbf{G}_\ell$ yields (30). ■

Proof of Lemma 1: According to the P – θ model, the modified active power injections under specification $\tilde{\mathbf{p}}$ and contingency ℓ follow the discrete-time affine system

$$\tilde{\mathbf{p}}_\ell^{t+1} = \tilde{\mathbf{G}}_\ell^\top \tilde{\mathbf{p}}_\ell^t + \tilde{\mathbf{p}}. \quad (48)$$

This recursion can be derived from (19) if we define $\tilde{\mathbf{p}}_\ell^t := \tilde{\mathbf{B}}_0 \tilde{\boldsymbol{\theta}}_\ell^t$ for all t . The goal is to characterize the set of vectors $\tilde{\mathbf{p}}_\ell^t$ for all contingencies ℓ , all iterations t , and every $\tilde{\mathbf{p}} \in \mathcal{C}_p$. To this end, unrolling (48) yields

$$\begin{aligned} \tilde{\mathbf{p}}_\ell^t &= \left(\mathbf{I} + \sum_{\tau=1}^t \tilde{\mathbf{G}}_\ell^\tau \right)^\top \tilde{\mathbf{p}} = \left(\mathbf{I} + \sum_{\tau=1}^t \beta_\ell^{\tau-1} \tilde{\mathbf{G}}_\ell \right)^\top \tilde{\mathbf{p}} \\ &= \left(\mathbf{I} + \frac{1 - \beta_\ell^t}{1 - \beta_\ell} \tilde{\mathbf{G}}_\ell \right)^\top \tilde{\mathbf{p}} = \beta_\ell^t \tilde{\mathbf{p}} + (1 - \beta_\ell^t) \tilde{\mathbf{M}}_\ell \tilde{\mathbf{p}}, \end{aligned} \quad (49)$$

where $\tilde{\mathbf{M}}_\ell := \mathbf{I}_N + \frac{1}{1 - \beta_\ell} \tilde{\mathbf{G}}_\ell^\top$. The second equality follows because $\tilde{\mathbf{G}}_\ell$ has a single non-zero eigenvalue $\beta_\ell \in (0, 1)$, and so $\tilde{\mathbf{G}}_\ell^\tau = \beta_\ell^{\tau-1} \tilde{\mathbf{G}}_\ell$; the third stems from the geometric series finite sum; and the fourth follows upon rearranging terms.

Equation (49) shows that $\tilde{\mathbf{p}}_\ell^t$ is a convex combination of $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{M}}_\ell \tilde{\mathbf{p}}$. Hence, for every $\tilde{\mathbf{p}} \in \mathcal{C}_p$, the training set $\mathcal{C}'_p$ should include the line segment between $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{M}}_\ell \tilde{\mathbf{p}}$ for all ℓ . Lemma 2 can be proved by exactly following the same procedure. ■

Proof of Lemma 3: We first add and subtract the vector $F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t))$ from $\hat{\mathbf{v}}_\ell^{t+1} - \mathbf{v}_\ell^{t+1}$. Applying the triangle inequality to the resultant vector norm provides

$$\begin{aligned} e_\ell^{t+1} &\leq \|F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\hat{\mathbf{v}}_\ell^t)) - F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t))\|_2 \\ &\quad + \|F_{\mathbf{w}}(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t)) - G_0(\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}_\ell^t))\|_2. \end{aligned}$$

The first term is bounded by $L_\ell e_\ell^t$ from (39). The second term is bounded by ϵ from (41) for all $\mathbf{c} + \mathbf{g}_\ell(\mathbf{v}) \in \mathcal{C}'$. Hence, the Euclidean distance between the two trajectories satisfies $e_\ell^{t+1} \leq L_\ell e_\ell^t + \epsilon$ for all t . Unfolding the iteration yields

$$e_\ell^t \leq L_\ell^t e_\ell^0 + \epsilon \sum_{\tau=0}^{t-1} L_\ell^\tau = L_\ell^t e_\ell^0 + \left(\frac{1 - L_\ell^t}{1 - L_\ell} \right) \epsilon.$$

If the two iterations are initialized at the same voltage point so that $e_\ell^0 = 0$, the claim in (44) follows since $L_\ell \in (0, 1)$. ■

REFERENCES

- [1] North American Electric Reliability Corporation, "NERC reliability standard PRC-003-0a," 2010, accessed: 2026-01-17. [Online]. Available: <https://www.nerc.com/pa/Stand/Reliability%20Standards/TPL-003-0a.pdf>
- [2] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power Generation, Operation, and Control*, 3rd ed. Hoboken, NJ, USA: Wiley, Dec. 2013.
- [3] T. Güler, G. Gross, and M. Liu, "Generalized line outage distribution factors," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 879–881, May 2007.
- [4] A. F. Castelli and J. D. Lara, "Improved evaluation of large network matrices for linear power flow within optimization problems," in *Proc. IEEE Power & Energy Society General Meeting*, Seattle, WA, Jul. 2024, pp. 1–5.
- [5] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2215–2224, Nov. 2012.
- [6] H. Qu, L. Roald, and G. Andersson, "Uncertainty margins for probabilistic AC security assessment," in *PowerTech*, Eindhoven, Netherlands, Jun. 2015, pp. 1–6.
- [7] S. V. Dhople, S. S. Guggilam, and Y. C. Chen, "Linear approximations to AC power flow in rectangular coordinates," in *Proc. Allerton Conf.*, Monticello, IL, Sep. 2015, pp. 211–217.
- [8] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, Nov. 2014.
- [9] J. Yu, Z. Li, J. Zhang, X. Bai, H. Ge, J. Zheng, and Q. H. Wu, "Efficient contingency analysis of power systems using linear power flow with generalized warm-start compensation," *International Journal of Electrical Power & Energy Systems*, vol. 156, p. 109692, Mar. 2024.
- [10] Y.-H. Yeung, A. Pothén, M. Halappanavar, and Z. Huang, "AMPS: An augmented matrix formulation for principal submatrix updates with application to power grids," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, pp. S809–S827, Oct. 2017.
- [11] R. Yao, F. Qiu, and K. Sun, "Contingency analysis based on partitioned and parallel holomorphic embedding," *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 1452–1462, Mar. 2022.
- [12] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *SmartGridComm*, Tempe, AZ, Nov. 2020, pp. 1–6.
- [13] X. Pan, T. Zhao, M. Chen, and S. Zhang, "DeepOPF: A deep neural network approach for security-constrained DC optimal power flow," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1725–1735, May 2021.
- [14] M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to solve the AC-OPF using sensitivity-informed deep neural networks," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2833–2846, Jul. 2022.
- [15] Y. Zhou, J. Park, and H. Zhu, "Scalable learning for optimal load shedding under power grid emergency operations," in *Proc. IEEE Power & Energy Society General Meeting*, Denver, CO, Jul. 2022, pp. 1–5.
- [16] T. Wu, A. Scaglione, S. Miguel, and D. Arnold, "Universal graph convolutional network: Zero-shot transfer across power grid topology reconfigurations," *IEEE Trans. Smart Grid*, pp. 1–1, May 2026.
- [17] S. Gupta, V. Kekatos, and S. Chatzivasileiadis, "Optimal design of volt/VAR control rules of inverters using deep learning," *IEEE Trans. Smart Grid*, vol. 15, no. 5, pp. 4731–4743, Sep. 2024.
- [18] W. Wang, N. Yu, J. Shi, and Y. Gao, "Safe off-policy deep reinforcement learning algorithm for volt/VAR control in power distribution systems," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3008–3018, Jul. 2020.
- [19] A. Kody, J. Pabon, and D. K. Molzahn, "Learning to solve the $N - k$ interdiction problem considering load variability," in *Hawaii International Conference on System Sciences*, Lahaina, HI, Jan. 2023, pp. 2723–2732.
- [20] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2082–2092, May 2021.
- [21] V. Eeckhout, H. Fani, M. U. Hashmi, and G. Deconinck, "Improved physics-informed neural network based AC power flow for distribution networks," in *Proc. IEEE Conf. on Innovative Smart Grid Technologies*, Dubrovnik, Croatia, Oct. 2024, pp. 1–6.
- [22] Y. Du, F. Li, J. Li, and T. Zheng, "Achieving 100x acceleration for $N - 1$ contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3303–3305, Jul. 2019.
- [23] A. M. Nakiganda, C. Cheylan, and S. Chatzivasileiadis, "Topology-aware neural networks for fast contingency analysis of power systems," *arXiv preprint arXiv:2310.04213*, Mar. 2025.
- [24] T. Wu, A. Scaglione, S. Miguel, and D. Arnold, "Universal graph convolutional network: Zero-shot transfer across power grid topology reconfigurations," *IEEE Trans. Smart Grid*, Jan. 2026.
- [25] N. Christianson, W. Cui, S. Low, W. Yang, and B. Zhang, "Fast and reliable $N - k$ contingency screening with input-convex neural networks," in *Learning for Dynamics and Control Conference*, vol. 283, Ann Arbor, MI, Jun. 2025, pp. 1–12, proceedings of Machine Learning Research.
- [26] M. Rahman, J. Qin, and V. Kekatos, "AC power flow contingency analysis using a single deep neural network," in *Hawaii Intl. Conf. on System Sciences (HICSS)*, Big Island, HI, Jan. 2027.
- [27] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fundamenta Mathematicae*, vol. 3, no. 1, pp. 133–181, 1922.
- [28] J. W. Simpson-Porco, "A theory of solvability for lossless power flow equations—part I: Fixed-point power flow," *IEEE Trans. Control of Network Systems*, vol. 5, no. 3, pp. 1361–1372, Sep. 2018.
- [29] M. Bazrafshan and N. Gatsis, "Convergence of the Z-bus method for three-phase distribution load-flow with ZIP loads," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 153–165, Jan. 2018.
- [30] S. Bolognani and S. Zampieri, "On the existence and linear approximation of the power flow solution in power distribution networks," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 163–172, Jan. 2016.
- [31] H. F. Walker and P. Ni, "Anderson acceleration for fixed-point iterations," *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715–1735, Aug. 2011.