Spatial Network Decomposition for Fast and Scalable AC-OPF Learning

Minas Chatzos, Terrence W.K. Mak, *Member, IEEE,* and Pascal Van Hentenryck, *Member, IEEE*

***Abstract*—This paper proposes a novel machine-learning approach for predicting AC-OPF solutions that features a fast and scalable training. It is motivated by the significant training time needed by existing machine-learning approaches for predicting AC-OPF. The proposed approach is a 2-stage methodology that exploits a spatial decomposition of the power network that is viewed as a set of regions. The first stage learns to predict the flows and voltages on the buses and lines coupling the regions, and the second stage trains, in parallel, the machine-learning models for each region. The predictions can then seed a power flow to eliminate the physical constraint violations, resulting in minor violations only for the operational bound constraints. Experimental results on the French transmission system (up to 6,700 buses) and large test cases from the `pglib` library (up to 9,000 buses) demonstrate the potential of the approach. Within a short training time, the approach predicts AC-OPF solutions with very high fidelity, producing significant improvements over the state-of-the-art. The proposed approach thus opens the possibility of training machine-learning models quickly to respond to changes in operating conditions.**

***Index Terms*—Optimal Power Flow; Machine Learning; Neural Networks; Network Decomposition;**

I. Introduction

The *AC Optimal Power Flow (AC-OPF)* problem is at the core of modern power system operations. It determines the least-cost generation dispatch that meets the demand of the power grid subject to engineering and physical constraints. It is non-convex and NP-hard [1], and the basic block of many applications, including security-constrained OPF [2], [3], security-constrained unit commitment [4], optimal transmission switching [5], capacitor placement [6], and expansion planning [7], among others.

Machine learning has significant potential for real-time AC-OPF applications for a variety of reasons [8]. A machine-learning model can leverage large amount of historical data and deliver extremely fast approximations (compared to an AC-OPF solver). Recent work (e.g., [9], [8]) has indeed shown that machine-learning approaches can predict AC-OPF with high fidelity and minimal constraint violations, using a combination of neural networks and Lagrangian duality. However, the training times and memory requirements of these machine-learning models can be quite significant, which limits their potential applications.

This paper explores a fundamentally different avenue: the design of a scalable machine-learning approach for predicting AC-OPF solutions that can be trained quickly. Such an approach would make it possible to train a machine-learning model quickly to accommodate new operating conditions. It would also open the possibility of training different machine-learning models for different time periods and to perform simulations with many more scenarios and contingencies.

To achieve this goal, the paper proposes a 2-stage machine-learning approach that exploits a spatial decomposition of the power system. The power network is viewed as a set of regions, the first stage learns to predict the flows and voltages on the buses and lines coupling the regions, and the second stage trains, in parallel, the machine-learning models for each region. Experimental results on the French transmission system (up to 6,700 buses and 9,000 lines) and other testcases (with more than 9,000 buses and 16,000 lines) demonstrate the potential of the approach. Within a short training time, the approach predicts AC-OPF solutions with very high fidelity and minor constraint violations, producing significant improvements over the state-of-the-art. Experimental results also show that the predictions can seed a power-flow optimization to return a solution within 0.05% of the AC-OPF objective, while reducing running times significantly.

To our knowledge, the proposed approach is the first *distributed* training algorithm for learning AC-OPF for large-scale network topology. It builds on top, and significantly extends, prior work [9], [8] combining machine learning and Lagrangian duality. Most importantly, the 2-stage approach significantly reduces the dimensionality of the learning task, allows the training to be performed in parallel for each region, and dramatically shortens training times, opening new avenues for machine learning in very large-scale system operations. It is also the first approach that can learn AC-OPF on an actual, large-scale tranmission system fast, even on reasonable hardware configurations. It is also important to emphasize that the proposed methodology is not restricted to AC-OPF and/or supervised learning. It should also be applicable to industrial Security-Constrained Economic Dispatch (SCED), or to other security-constrained forrmulations. It can also be applied to other learning approaches, such as reinforcement learning methods based on neural networks for real-time OPF that uses the existing state of the system.

II. Related Work

Machine learning has attracted significant attention in the power systems community: recent overviews of the various approaches and applications can be found in [10], [11]. The research on the AC-OPF can be classified into two categories: supervised and reinforcement learning. On the supervised

The authors are affiliated with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. E-mail: minas@gatech.edu, wmak@gatech.edu, pvh@isye.gatech.edu.

learning side, several approaches have been proposed for learning the active set of constraints [12], [13], [14], [15], [16], imitating the Newton-Raphson algorithm [17], or learning warm-start points for speeding-up the optimization process [18], [19]. Several approaches aim to predict optimal dispatch decisions [20], [21], [22] but these were limited to small-case studies. As mentioned earlier, this paper expands the work from [9], [8] which has shown how deep learning can predict AC-OPF for large test cases with high fidelity and minimal constraint violations, using a combination of neural networks and Lagrangian duality. [9] also showed how to exploit prior solution or the system state and mentioned that the predictions can be used to replace existing approximations that seed a power flow. In the context of DC-OPF, it is worth mentioning the results of [23], [24] that provide formal guarantees on the predictions of neural networks. The application of machine learning to the security-constrained extension of the DC-OPF is presented in [25], [26]. Note that these prior works all train the model in a centralized fashion. The main contribution of this paper is a distributed machine-learning scheme which is fast and scalable. Several reinforcement-learning approaches have also been proposed for the OPF [27], [28], [29], [30]. These approaches, which focus on solving real-time AC-OPF, also use DNNs for approximating a mapping between the state (loads) and the actions (generator, voltage setpoints) of the agents, and their performance has been reported on small topologies (up to 200 buses). The 2-stage methodology proposed in this paper is particular intriguing in that context, since it can boost these approaches by exploiting the spatial and physical properties of the power system.

III. Preliminaries

a) The AC Optimal Power Flow Problem: A power network is modeled as an undirected graph $(\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ and $\mathcal{E}$ are the set of buses and transmission lines. The set of generators and loads are denoted by $\mathcal{G}$ and $\mathcal{L}$. The goal of the OPF is to determine the generator dispatch of minimal cost that satisfies the load. The OPF constraints include engineering and physical constraints. The OPF formulation is shown in Figure 1. The power flow equations are expressed in terms of complex power of the form $S=(p+jq)$, where p and q denote the active and reactive powers, admittances of the form $Y=(g+jb)$, where g and b denote the conductance and susceptance, and voltages of the form $V=(v\angle\theta)$, with magnitude v and phase angle θ. The formulation uses v_i, θ_i, p_i^g, and q_i^g to denote the voltage magnitude, phase angle, active power generation, reactive power generation at bus i. Moreover, p_{ij}^f and q_{ij}^f denote the active and reactive power flows associated with line (i,j). The set S_i represents the set of shunts in bus i. The OPF receives as input the demand vectors p_i^d and q_i^d for each bus i. The objective function captures the cost of the generator dispatch. Typically, $c_i(\cdot)$ is a linear or quadratic function. Constraints (2), (3r), and (3i) capture operating bounds for the associated variables. The thermal limit for line (i,j) is captured via constraint (4). Constraints (5r) and (5i) capture *Ohm's Law*. Branch shunts are also considered in the experimental results, but omitted from the formulation for simplicity. Constraints (6r) and (6i) capture *Kirchhoff's Law*.

$$\textbf{minimize} \quad \sum_{i=1}^{|\mathcal{N}|} c_i(p_i^g) \tag{1}$$

subject to:

$$\underline{v}_i \le v_i \le \overline{v}_i \qquad \forall i \in \mathcal{N} \tag{2}$$

$$\underline{p}_i^g \le p_i^g \le \overline{p}_i^g \qquad \forall i \in \mathcal{N} \tag{3r}$$

$$\underline{q}_i^g \le q_i^g \le \overline{q}_i^g \qquad \forall i \in \mathcal{N} \tag{3i}$$

$$(p_{ij}^f)^2 + (q_{ij}^f)^2 \le |\overline{S}|_{ij}^2 \qquad \forall (i,j) \in \mathcal{E} \tag{4}$$

$$p_{ij}^f = g_{ij} v_i^2 - v_i v_j (b_{ij} \sin(\theta_i - \theta_j) + g_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (i,j) \in \mathcal{E} \tag{5r}$$

$$q_{ij}^f = -b_{ij} v_i^2 - v_i v_j (g_{ij} \sin(\theta_i - \theta_j) - b_{ij} \cos(\theta_i - \theta_j)) \qquad \forall (i,j) \in \mathcal{E} \tag{5i}$$

$$p_i^g - p_i^d - v_i^2 \sum_{k \in S_i} g_k = \sum_{(i,j) \in \mathcal{E}} p_{ij}^f \qquad \forall i \in \mathcal{N} \tag{6r}$$

$$q_i^g - q_i^d + v_i^2 \sum_{k \in S_i} b_k = \sum_{(i,j) \in \mathcal{E}} q_{ij}^f \qquad \forall i \in \mathcal{N} \tag{6i}$$

Fig. 1: The OPF Formulation.

b) Neural Network Architectures: Neural networks have achieved tremendous success in approximating highly complex, nonlinear mappings in various domains and applications. A Neural Network (NN) consists of a series of layers, the output of each layer being the input to the next layer. The NN layers are often fully connected and the function connecting the layers is given by $\boldsymbol{o} = \pi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{x} \in \mathbb{R}^n$ is the input vector, $\boldsymbol{o} \in \mathbb{R}^m$ the output vector, $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ a weight matrix, and $\boldsymbol{b} \in \mathbb{R}^m$ a bias vector. The function $\pi(\cdot)$ is non-linear (e.g., a rectified linear unit (ReLU)).

c) Notations: The cardinality of set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. $[N]$ represents the set $\{1, 2, \ldots, N\}$. Vectors are displayed using bold letters and $\boldsymbol{x} = [x_1, x_2, ..., x_n]^\top$. The element-wise lower (resp. upper) bound of the vector $\boldsymbol{x}$ is denoted by $\underline{\boldsymbol{x}}$ (resp. $\overline{\boldsymbol{x}}$). In learning algorithms, the prediction for $\boldsymbol{x}$ is denoted by $\hat{\boldsymbol{x}}$.

IV. Learning AC-OPF

A. OPF Learning Goals

Given loads $(\boldsymbol{p^d}, \boldsymbol{q^d})$, the learning goal is to predict the optimal control setpoints $(\boldsymbol{p^g}, \boldsymbol{q^g})$ of the generators, the bus voltage $\boldsymbol{v}$, and the phase angle difference $\boldsymbol{\Delta\theta}$ of the lines. This task is equivalent to learning the complex, nonlinear, high-dimensional mapping:

$$\mathcal{O} : \mathbb{R}^{2|\mathcal{L}|} \to \mathbb{R}^{|\mathcal{N}|+|\mathcal{E}|+2|\mathcal{G}|} \tag{7}$$

which maps the loads onto the optimal AC-OPF solution returned by a deterministic solver. The input to the learning task is a dataset

$$\mathcal{D} = \{(\boldsymbol{p^d}, \boldsymbol{q^d})^t, (\boldsymbol{v}, \boldsymbol{\Delta\theta}, \boldsymbol{p^g}, \boldsymbol{q^g})^t\}_{t=1}^T$$

consisting of T instances specifying the inputs and outputs.

B. A Lagrangian Dual Model for Learning AC-OPF

One of the challenges of learning mapping $\mathcal{O}$ is the presence of physical and engineering constraints. Ideally, given a NN

$\mathcal{O}[\boldsymbol{w}]$ parameterized by weights $\boldsymbol{w}$, the goal is to find the optimal solution $\boldsymbol{w}^*$ of the problem:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \mathbb{L}_0(\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}) \\ \text{s.t.} \quad & (\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}) = \mathcal{O}[\boldsymbol{w}](\boldsymbol{p}^{\boldsymbol{d}}, \boldsymbol{q}^{\boldsymbol{d}}) \\ & (\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}, \hat{\boldsymbol{p}}^{\boldsymbol{f}}, \hat{\boldsymbol{q}}^{\boldsymbol{f}}) \text{ satisfy (2)-(6}i\text{)} \end{aligned} \tag{10}$$

where $\mathbb{L}_0$ denotes the average norm of the difference between the ground truth and the predictions $\mathbb{L}_0(\hat{\boldsymbol{x}}) = \frac{1}{T}\sum_{t=1}^{T} ||\boldsymbol{x}^t - \hat{\boldsymbol{x}}^t||$ over all training instances, and $(\hat{\boldsymbol{p}}^{\boldsymbol{f}}, \hat{\boldsymbol{q}}^{\boldsymbol{f}})$ are computed using constraints (5r) and (5i). However, it is unlikely that there exist weights $\boldsymbol{w}$ such that the predictions actually satisfy the AC-OPF constraints, since the learning task is a high-dimensional regression task. However, ignoring the constraints entirely leads to predictions that significantly violate the problem constraints as shown in [8], [26]. The approach from [9], [8] addresses this difficulty by using a Lagrangian dual method relying on constraint violations. The violation of a constraint $f(x) \geq 0$ is given by $\nu_c(x) = \max\{0, -f(x)\}$, while the violation of $f(x) = 0$ is $\nu_c(x) = |f(x)|$. Problem (10) can then be approximated by

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \mathbb{L}(\boldsymbol{\lambda}, \boldsymbol{w}) = \mathbb{L}_0(\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}) + \boldsymbol{\lambda}^\top \bar{\boldsymbol{\nu}} \\ \text{s.t.} \quad & (\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}) = \mathcal{O}[\boldsymbol{w}](\boldsymbol{p}^{\boldsymbol{d}}, \boldsymbol{q}^{\boldsymbol{d}}) \end{aligned} \tag{11}$$

where $\boldsymbol{\lambda}^\top \bar{\boldsymbol{\nu}} = \sum_{c\in\mathbb{C}} \lambda_c \bar{\nu}_c(\hat{\boldsymbol{v}}, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}})$, λ_c is the weight for the violation of constraint c, and $\bar{\nu}_c$ denotes the average violation of constraint c over all training instances. Again, the satisfaction of the constraints (5r), (5i) is guaranteed, since the power flows are computed indirectly from these constraints. For a fixed $\boldsymbol{\lambda}$, $\mathbb{L}(\boldsymbol{\lambda}, \boldsymbol{w})$ can be used as the loss function for training the neural network. Moreover, the constraint weights can be updated using a subgradient method that performs the following operations in iteration j.

$$\begin{aligned} \boldsymbol{w}^j &= \arg\min_{\boldsymbol{w}} \mathbb{L}(\boldsymbol{\lambda}^{(j-1)}, \boldsymbol{w}) \\ \boldsymbol{\lambda}^j &= \boldsymbol{\lambda}^{(j-1)} + \rho \bar{\boldsymbol{\nu}}(\boldsymbol{w}^j) \end{aligned} \tag{12}$$

Learning the mapping $\mathcal{O}$ is challenging for large-scale topologies. For instance, it takes 7 hours to train a network for a topology of 3500 buses [8]. This limits the potential applications of neural networks in large power systems which may be up to 50000 buses. Indeed, during operations, the topology of the system may change from day to day through line or bus switching, meaning that a different mapping needs to be learned. Similarly, the mapping $\mathcal{O}$ depends on the commitment decisions in the day-ahead markets, again potentially changing the mapping to be learned.

The goal of this paper is to propose *a fast training procedure to learn the mapping* $\mathcal{O}$. Such a fast training procedure would have many advantages: the NN model could be trained after the day-ahead market clearing and/or in real time during operations when the network topology changes, and it could be tailored to the load profiles of specific times in the day (e.g., 2:00pm-4:00pm). These considerations are important, especially given the increasing share of renewable energy in the energy mix and the increasing prediction errors. For instance, a fast training procedure enables machine-learning models to be trained and refined at various times during the day when forecasts on renewable energy sources are becoming increasingly reliable.

K	$\max_{k\in[K]} \lvert\mathcal{N}^k\rvert$	$\lvert\mathcal{E}^{\leftrightarrow}\rvert$
3	2525	127
6	1653	279
12	1156	326

TABLE I: Maximum Region Sizes and Number of Coupling Branches for Different Partitions of the French System.

C. Exploiting Network Sparsity

One possible avenue to obtain a fast training procedure is to exploit the sparsity typically found in power system networks. Consider a partition $\{\mathcal{N}^k\}_{k=1}^K$ of the buses, i.e.,

$$\bigcup_{k=1}^{K} \mathcal{N}^k = \mathcal{N}, \quad \mathcal{N}^k \cap \mathcal{N}^{k'} = \emptyset, k \neq k'$$

Denote the generators and loads of region k by $\mathcal{G}^k$ and $\mathcal{L}^k$ respectively and define

$$\mathcal{E}^k = \{(i,j) \in \mathcal{E} : i, j \in \mathcal{N}^k\}, k \in [K].$$

$$\mathcal{E}^{\leftrightarrow} = \mathcal{E} \backslash (\cup_{k=1}^{K} \mathcal{E}^k), \quad \mathcal{N}^{\leftrightarrow} = \{i : (i,j) \in \mathcal{E}^{\leftrightarrow} \vee (j,i) \in \mathcal{E}^{\leftrightarrow}\}$$

Here $\mathcal{E}^k$ represents the lines within partition element k and $\mathcal{E}^{\leftrightarrow}$ the coupling lines that connect partition elements. In the French transmission system, the test case in this paper, $|\mathcal{N}| = 6705$, $|\mathcal{E}| = 8962$, and $|\mathcal{E}| \approx 1.3\times|\mathcal{N}|$. Moreover, the system is organized in 12 geographical areas using 326 (3.6%) coupling lines and $\max_{k\in[K]} |\mathcal{N}^k| = 1156$ (17.2%) buses. The trade-off between the maximum region size and the number of coupling branches is illustrated in Table I.

To leverage the network sparsity, a natural first attempt would be to learn a mapping for each region, i.e.,

$$\mathcal{O}_0^k : \mathbb{R}^{2|\mathcal{L}^k|} \to \mathbb{R}^{2|\mathcal{N}^k|+2|\mathcal{G}^k|} \quad (k \in [K]). \tag{8}$$

The learning thus predicts the setpoints for generators in region k using only the loads of the same region. These learning tasks would be performed independently and in parallel. However, it is obvious that the loads $\mathcal{L}^k$ are not sufficient to determine the optimal setpoints for generators $\mathcal{G}^k$. In fact, $\mathcal{O}_0^k$ is not even a function, since two inputs for the loads $\mathcal{L}^k$ in the training set may be associated with different outputs due to loads in other parts of the network.

D. Capturing Flows on Coupling Lines

Consider the simplistic power system depicted in Figure 2. There are two areas, $\mathcal{N}^1 = \{1,2,3\}$ and $\mathcal{N}^2 = \{4,5,6\}$, which gives $\mathcal{N}^{\leftrightarrow} = \{3,4\}$, $\mathcal{E}^{\leftrightarrow} = \{(3,4),(4,3)\}$. The mapping $\mathcal{O}$ views the setpoints for the generator at bus 2 (p_2^g, q_2^g) as a function of $(p_1^d, q_1^d, p_4^d, q_4^d, p_5^d, q_5^d, p_6^d, q_6^d)$. However, assume that flows $(p_{4,3}^f, q_{4,3}^f)$, along with the voltage magnitudes v_3, v_4 are fixed and respect constraints (4), (5r), (5i) associated with line $(3,4)$. In that case, the setpoints for the generator at bus 2 can be computed without the knowledge

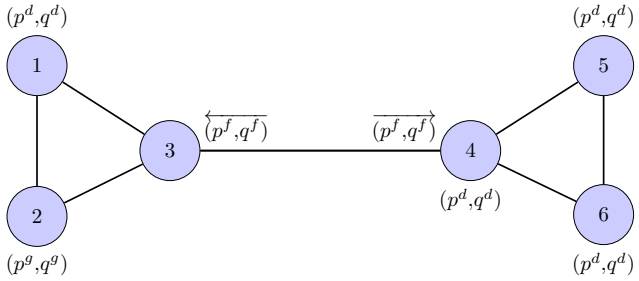


Fig. 2: A Simple Network With $|\mathcal{N}| = 6$, $|\mathcal{E}| = 7$, $|\mathcal{G}| = 1$, and $|\mathcal{L}| = 4$.

of $(p_4^d, q_4^d, p_5^d, q_5^d, p_6^d, q_6^d)$: the vector $(p_{4,3}^f, q_{4,3}^f, v_3)$ encodes all the information from area 2 needed to compute the generator setpoint. Hence, one may attempt to express (p_2^g, q_2^g) as a function of $(p_1^d, q_1^d, p_{4,3}^f, q_{4,3}^f, v_3)$ which decreases the input size from 8 to 5. The input size decreases by 3 in this example but the size reduction is significantly larger in actual systems.

With this in mind, the mappings in Equation 8 become

$$\mathcal{O}^k : \mathbb{R}^{2|\mathcal{L}^k|+|\mathcal{N}^{\to k}|+2|\mathcal{E}^{\to k}|} \to \mathbb{R}^{|\mathcal{N}^k \setminus \mathcal{N}^{\to k}|+|\mathcal{E}^k|+2|\mathcal{G}^k|} \quad (9)$$

where the coupling lines, buses of region k are defined as

$$\mathcal{E}^{\to k} = \{(i,j) \in \mathcal{E}^{\leftrightarrow} : i \in \mathcal{N}^k \ \vee j \in \mathcal{N}^k\}$$
$$\mathcal{N}^{\to k} = \mathcal{N}^k \cap \mathcal{N}^{\leftrightarrow}$$

$\mathcal{O}^k$ maps the loads in area k, the flows to area k, and the voltage of the coupling buses to the optimal generator setpoints in the area, i.e., the active and reactive outputs of the regional generators, the phase angle differences of the regional branches, and the voltage setpoints for the non-coupling buses of the region. For large transmission systems, the input/output dimensions of each mapping $\mathcal{O}^k$ are significantly smaller that those of $\mathcal{O}$. The learning tasks can proceed in parallel and their complexity is reduced, since each mappings $\mathcal{O}^k$ is an order of magnitude smaller in size than $\mathcal{O}$.

Unfortunately, this approach has a key limitation: *each mapping $\mathcal{O}^k$ can be learned from historical data but cannot be used for prediction since the coupling flows and voltages are not known at prediction time*. Indeed, during training, the learning task has access to the coupling values for each instance. However, this is not true at prediction time. The next section shows how to overcome this difficulty.

V. Two-Stage Learning of AC-OPF

The fast training method for AC-OPF is a two-stage approach: the first stage is a NN that predicts the flow on the coupling lines and the second stage is a collection of NNs, each of which approximates a mapping $\mathcal{O}^k$.

A. Learning Coupling Voltages & Flows

The goal of the first stage is to learn the mapping

$$\mathcal{O}^{\leftrightarrow} : \mathbb{R}^{2|\mathcal{L}|} \to \mathbb{R}^{|\mathcal{N}^{\leftrightarrow}|+|\mathcal{E}^{\leftrightarrow}|} \quad (13)$$

Algorithm 1: The First-Stage Coupling Training.

$\lambda_c \leftarrow 0, \forall c \in \mathbb{C}^{\leftrightarrow}$
for $i = 1, 2, ..., epochs_\lambda$ **do**
  for $j = 1, 2, ..., epochs_w$ **do**
    $(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0) \leftarrow \mathcal{O}^{\leftrightarrow}[\boldsymbol{w}^0](\boldsymbol{p^d}, \boldsymbol{q^d})$
    $\mathbb{L}^{\leftrightarrow}(\boldsymbol{\lambda}, \boldsymbol{w}^0) \leftarrow \mathbb{L}_0(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0) + \sum_{c\in\mathbb{C}^{\leftrightarrow}} \lambda_c \bar{\nu}_c(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0)$
    $\boldsymbol{w}^0 \leftarrow \boldsymbol{w}^0 - \alpha\nabla_{\boldsymbol{w}^0}(\mathbb{L}^{\leftrightarrow}(\boldsymbol{\lambda}, \boldsymbol{w}^0))$
  end
  $\lambda_c \leftarrow \lambda_c + \rho\bar{\nu}_c, \quad \forall c \in \mathbb{C}^{\leftrightarrow}$
end
Result: Weights $(\boldsymbol{w}^0)^*$

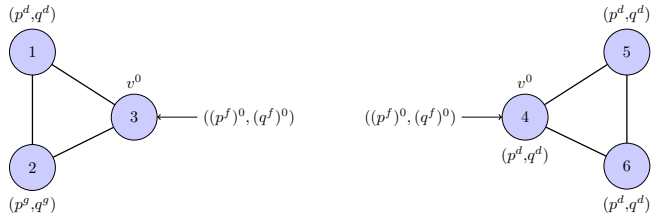


Fig. 3: Illustration of the Decomposition.

from the loads to the voltages magnitude $\boldsymbol{v}^0$ of the coupling buses and the phase angle difference $\boldsymbol{\Delta\theta}^0$ of the coupling branches. Although the mapping considers all loads, it can be learned fast (e.g., under 30 minutes) even for large networks, because of the small number of coupling buses and lines. The coupling flows are then computed indirectly via constraints (5r) and (5i). Let $\mathbb{C}^{\leftrightarrow}$ denote the set of constraints (2) for $i \in \mathcal{N}^{\leftrightarrow}$, and (4) and (5$r$), (5$i$) for $(i,j) \in \mathcal{E}^{\leftrightarrow}$. The learning task uses a neural network $\mathcal{O}^{\leftrightarrow}[\boldsymbol{w}^0]$ parameterized by weights $\boldsymbol{w}^0$ and predicts the coupling voltages $(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0)$. The loss function is given by:

$$\mathbb{L}^{\leftrightarrow}(\boldsymbol{\lambda}, \boldsymbol{w}^0) = \mathbb{L}_0(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0) + \sum_{c\in\mathbb{C}^{\leftrightarrow}} \lambda_c \bar{\nu}_c(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0)$$

The training follows equation (12) and the resulting optimal weights $(\boldsymbol{w}^0)^*$ lead to the first-stage predictions

$$(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0) = \mathcal{O}^{\leftrightarrow}[(\boldsymbol{w}^0)^*](\boldsymbol{p^d}, \boldsymbol{q^d})$$

and the resulting first-stage coupling flow predictions $(\hat{\boldsymbol{p}}^{\boldsymbol{f}})^0, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^0$. The first stage is summarized in Algorithm 1.

B. Training of Regional Systems

The training of the regional systems uses the first-stage predictions for the coupling flows and voltages. Note however that it could use the ground truth present in the instance data, but experimental results have shown that this degrades the overall prediction accuracy. The decoupling is illustrated in Figure 3, where the voltages of the coupling buses 3 and 4 and the incoming/outgoing flows for each region are fixed to the first-stage predictions.

To learn mappings $\mathcal{O}^k$ $(k \in [K])$, let $\mathbb{C}^k$ denote the set of constraints associated with region k, i.e., constraint (2) for buses $i \in \mathcal{N}^k \setminus \mathcal{N}^{\to k}$, constraints (3$r$), (3$i$), (6$r$), and (6$i$) for buses $i \in \mathcal{N}^k$, and constraints (4), (5r), (5i) for branches

Algorithm 2: The Second-Stage Training For Sub-Network k.

1 $\lambda_c \leftarrow 0, \forall c \in \mathbb{C}^k$
2 $(\hat{\boldsymbol{v}}^0, \widehat{\boldsymbol{\Delta\theta}}^0) \leftarrow \mathcal{O}^{\leftrightarrow}[(\boldsymbol{w}^0)^*](\boldsymbol{p^d}, \boldsymbol{q^d})$
3 Compute $(\hat{\boldsymbol{p}}^{\boldsymbol{f}})^{0,k}, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^{0,k}$ via (5r), (5i)
4 **for** $i = 1, 2, ..., epochs_\lambda$ **do**
5 **for** $j = 1, 2, ..., epochs_w$ **do**
6 $(\hat{\boldsymbol{v}}^k, \widehat{\boldsymbol{\Delta\theta}}^k, (\hat{\boldsymbol{p}}^{\boldsymbol{g}})^k, (\hat{\boldsymbol{q}}^{\boldsymbol{g}})^k) \leftarrow \mathcal{O}^k[\boldsymbol{w}^k]((\boldsymbol{p^d})^k, (\boldsymbol{q^d})^k, \hat{\boldsymbol{v}}^{0,k}, (\hat{\boldsymbol{p}}^{\boldsymbol{f}})^{0,k}, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^{0,k})$
7 $\mathbb{L}^k(\boldsymbol{\lambda}, \boldsymbol{w}^k) \leftarrow \mathbb{L}_0(\hat{\boldsymbol{v}}^k, \widehat{\boldsymbol{\Delta\theta}}^k, (\hat{\boldsymbol{p}}^{\boldsymbol{g}})^k, (\hat{\boldsymbol{q}}^{\boldsymbol{g}})^k) + \sum_{c\in\mathbb{C}^k} \lambda_c \bar{\nu}_c((\hat{\boldsymbol{v}}, \widehat{\boldsymbol{\Delta\theta}}, \hat{\boldsymbol{p}}^{\boldsymbol{g}}, \hat{\boldsymbol{q}}^{\boldsymbol{g}}, \hat{\boldsymbol{v}}^0, (\hat{\boldsymbol{p}}^{\boldsymbol{f}})^0, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^0)^k)$
8 $\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \alpha \nabla_{\boldsymbol{w}^k}(\mathbb{L}^k(\boldsymbol{\lambda}, \boldsymbol{w}^k))$
9 **end**
10 $\lambda_c \leftarrow \lambda_c + \rho \bar{\nu}_c, \quad \forall c \in \mathbb{C}^k$
11 **end**
Result: Weights $(\boldsymbol{w}^k)^*$

$(i,j) \in \mathcal{E}^k$. In particular, the power balance constraint (6r), (6i) for region k becomes

$$p_i^g - (p_i^d - \sum_{(i,j)\in\mathcal{E}^{\rightarrow k}} (p_{ij}^f)^0) = \sum_{(i,j)\in\mathcal{E}^k} p_{ij}^f \qquad i \in \mathcal{N}^k$$
$$q_i^g - (q_i^d - \sum_{(i,j)\in\mathcal{E}^{\rightarrow k}} (q_{ij}^f)^0) = \sum_{(i,j)\in\mathcal{E}^k} q_{ij}^f \qquad i \in \mathcal{N}^k$$

The learning task uses a collection $\{\mathcal{O}^k[\boldsymbol{w}^k]\}_{k\in[K]}$ of NNs and the loss function for each regional net is given by

$$\mathbb{L}^k(\boldsymbol{\lambda}, \boldsymbol{w}^k) = \mathbb{L}_0(\hat{\boldsymbol{v}}^k, \widehat{\boldsymbol{\Delta\theta}}^k, (\hat{\boldsymbol{p}}^{\boldsymbol{g}})^k, (\hat{\boldsymbol{q}}^{\boldsymbol{g}})^k) + \sum_{c\in\mathbb{C}^k} \lambda_c \bar{\nu}_c(\hat{\boldsymbol{v}}^k, \widehat{\boldsymbol{\Delta\theta}}^k, (\hat{\boldsymbol{p}}^{\boldsymbol{g}})^k, (\hat{\boldsymbol{q}}^{\boldsymbol{g}})^k, \hat{\boldsymbol{v}}^{0,k}, (\hat{\boldsymbol{p}}^{\boldsymbol{f}})^{0,k}, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^{0,k})$$

where $\hat{\boldsymbol{v}}^{0,k}$ is the first-stage prediction for the voltage magnitude of the coupling buses of region k and $(\hat{\boldsymbol{p}}^{\boldsymbol{f}})^{0,k}, (\hat{\boldsymbol{q}}^{\boldsymbol{f}})^{0,k}$ the first-stage predictions for the incoming/outgoing flows of region k. The training, summarized in Algorithm 2, is performed using the approach in equation (12) and each region can be trained in parallel. Line 2 predicts the voltage setpoints for the coupling buses and the phase angle differences of the coupling lines. Line 3 computes the predicted coupling flows from these predictions. Line 6 computes the predictions for region k given the current NN parameters and constraint weights. Line 8 performs the back-propagation to update the weights and line 10 updates the constraint weights.

C. Feasibility Restoration

Since the proposed learning method is a regression task, its predictions will violate some constraints. This section proposes two methods to obtain AC-feasible solutions.

1) Projection onto the AC-Feasible Set: This first method, denoted by PROJ, is an optimization model to project the prediction onto its closest feasible AC-OPF solution, i.e.,

$$\begin{aligned} \min \quad & ||\boldsymbol{p^g} - \hat{\boldsymbol{p}}^{\boldsymbol{g}}||_2^2 + ||\boldsymbol{v} - \hat{\boldsymbol{v}}||_2^2 \\ \text{s.t} \quad & (2) \text{ - } (6i) \end{aligned}$$

Find $\boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{p^g}, \boldsymbol{q^g}$
subject to:

$$\begin{aligned} & v_i = \max(\min(\hat{v}_i, \overline{v}_i), \underline{v}_i), \theta_i = 0, && \forall i \in \mathcal{N}_1 \\ & p_i^g = 0, q_i^g = 0, && \forall i \in \mathcal{N}_2 \\ & v_i = \max(\min(\hat{v}_i, \overline{v}_i), \underline{v}_i), p_i^g = \max(\min(\hat{p}_i^g, \overline{p}_i^g), \underline{p}_i^g), && \forall i \in \mathcal{N}_3 \\ & (5r), (5i), (6r), (6i) \end{aligned}$$

Fig. 4: The `Power-Flow` Formulation.

2) Power Flow: The second method follows the approach used in practice and applies the ubiquitous Power Flow (PF) to the predictions. See, for instance, [31][Chapter 6] for a detailed presentation of the power flow problem. PF partitions the buses of the system in three categories, i.e., the slack bus ($\mathcal{N}_1$), the P-Q buses ($\mathcal{N}_2$), and the P-V buses ($\mathcal{N}_3$), and fixes two variables at each bus depending on its type. The Power-Flow model is a system of $2|\mathcal{N}|$ equations depicted in Figure 4 and the predictions are used to fix variables p_i^g, v_i for the P-V buses. Since these predictions may have slight bound violations, they are clamped between their lower and upper bounds. PF ignores the operational bound constraints and the cost of the dispatch, but it returns a solution that satisfies the physical constraints of the system.

Power flow algorithms used in practice are substantially more involved than the model shown in Figure 4 and excutes multiple phases. It is outside the scope of the paper to reproduce them. However, to improve the satisfaction of the bound constraints, the experimental results use a two-phase approach. Since the constraint violations arising when solving PF mostly concern reactive power bounds, a second PF phase frees the voltage variables from the PV buses and transforms them into P-Q buses using the reactive power predictions.

VI. EXPERIMENTAL RESULTS

This section presents the core experimental results. Section IX-A in the appendix discusses the sensitivity of the models to the size of the training dataset, and Section IX-B illustrates how the proposed approach can be used to train a machine-learning model quickly when operating conditions change.

A. Experimental Setting

The AC-OPF, PROJ, and PF optimization models were solved in a centralized fashion using the under JuMP package for with Julia 1.5.4 with the nonlinear solver IPOPT [32] and with the linear solver MA57 and tolerance 10^{-6}. The linear solver MA57 significantly decreases solving times compared to the default linear solver. The configuration uses 2.5 GHz-i7 Intel Cores and 16GB of RAM. In total, 10^4 load profiles, which correspond to feasible AC-OPF problems, were generated for each test case. 80% of these instances were used for training and the remaining 20% for testing. The learning models were implemented using PyTorch [33] and trained using NVidia Tesla V100 GPUs with 16GB of memory. The Pytorch package automatically computes gradients. The training of each network utilizes mini-batches of size 120 and

the learning rate α was set to be decreasing from 10^{-3} to 10^{-6}, while ρ was set to 10^{-3}. All the data utilized by the learning and optimization procedures are in per-unit.

This section includes a comparison between model $\mathbb{O}$ that directly approximates mapping $\mathcal{O}$ with the proposed two-stage approach $\mathbb{D}$. The first-stage learning model of $\mathbb{D}$ is a NN with two fully connected subnetworks with ReLU activation with sizes $2|\mathcal{L}|\times|\mathcal{N}^{\leftrightarrow}|\times|\mathcal{N}^{\leftrightarrow}|$ and $2|\mathcal{L}|\times|\mathcal{E}^{\leftrightarrow}|\times|\mathcal{E}^{\leftrightarrow}|$ for the voltage magnitudes and phase angle differences respectively. For each region, each NN topology consists of 4 fully connected subnetworks with ReLU activation, one for each predicted variable. The subnetworks have one hidden layer and size $2|\mathcal{L}^k|\times 3|\mathcal{L}^k|\times|\mathcal{G}^k|$. Model $\mathbb{O}$ is similar in structure and has four fully-connected subnetworks of size $2|\mathcal{L}|\times 3|\mathcal{L}|\times|\mathcal{G}|$. The learning models were allocated 90 minutes of training time. Model $\mathbb{O}$ was trained using the centralized equivalent of Algorithm 2 which was presented in [8]. For $\mathbb{D}$, 30 minutes is allocated to the first stage, and 60 minutes to the second stage. The training window does not necessarily need to be 90 minutes. Depending on the operational needs, larger or smaller windows may be considered.

B. Load Profiles

The test cases (Table II) are parts of the actual French Transmission System. `France` is the French transmission system, `France_EHV` is the very high-voltage French system, and `France_LYON` is `France_EHV` with a detailed representation of the Lyon region. The French system is organized in 12 geographical regions. The dataset is generated by taking into account this geographical information. A load l in region k with nominal value $(p^d, q^d)^0$ is generated by

$$(p^d, q^d) = (\alpha + \beta^k + \gamma^l)((p^d)^0, (q^d)^0)$$

where the following coefficients are randomly drawn from the following distributions

$$\begin{aligned} \alpha &\sim \text{Uniform}[0.875, 0.975] \\ \beta^k &\sim \text{Uniform}[-0.025, 0.025], \quad \forall k \in [K] \\ \gamma^l &\sim \text{Uniform}[-0.0025, 0.0025], \quad \forall l \in \mathcal{L} \end{aligned}$$

The term α captures the system-wide load level, while β^k is associated with differences in the loads between regions (e.g., due to potentially different weather conditions). The difference in coefficients may be up to 5% for two different regions. Finally, γ^l is the uncorrelated noise added to each individual load with a range of 0.5% of its nominal value.

The resulting dataset captures realistic load profiles: the uniform load perturbation, the load level differences between the regions, and the fixed active, reactive power ratio represent the typical behavior for aggregated demand in a large-scale topology spanning several geographic regions. Randomly perturbing each individual load in an uncorrelated fashion would produce unrealistic load profiles: they would lead to an unnecessarily challenging learning task that would need to capture an exponential number of unrealistic behaviors of the power system. To highlight this point, Figure 5 depicts the actual consumption for three French regions over a 12 hour interval. Observe the strong correlation of the demand between the three regions. However, the correlation is not perfect and the ratios between the regional loads vary by small factors. The term β_k is used to account for this behavior. The resulting load profiles range from 0.85 of the nominal to the nominal load. This 15% difference is typical over a 12-hour interval as shown in Figure 5.

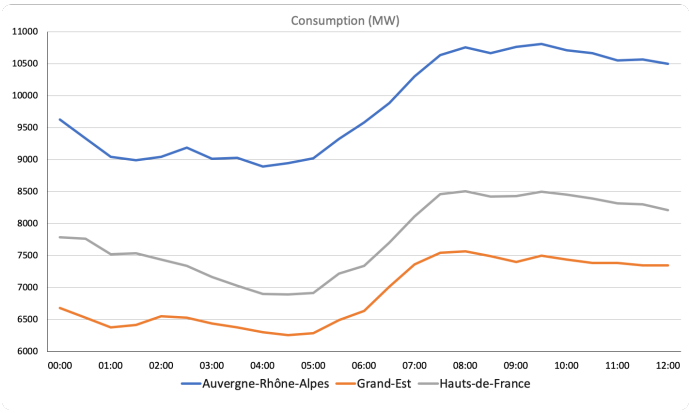


Fig. 5: Consumption for Three Regions in the French System over a 12-hour Interval.

C. First-stage Predictions

This section presents the prediction errors of the first stage. The training time was limited to 30 minutes. Table III contains aggregate results for the active and reactive powers of the coupling branches, as well the voltage magnitudes, for all three test cases. The results are an average over all instances and coupling branches. The average error is close to 1 MW for the largest two test cases: `France` and `France_Lyon`. Meanwhile, the 95-Quantile indicates that 95% of the predictions result in an error less that 5 MW. In the smaller `France_EHV`, the errors are slightly higher reaching 3.5 MW on average. Given that the nominal load of the `France` system is $50,000$ MW) and the nominal flow values are greater than 100 MW and up go to $1,000$ MW), these results indicate that the prediction errors are small in percentage for all test cases. Table III also shows that the voltage magnitudes are predicted very accurately. Figure 6 contains detailed results on the active part of the flows of the coupling branches for the `France` test case, showing consistent results across all tested instances. The 95% quantile graph indicates that the prediction errors exceed 5 MW only for a very small percentage of the test cases and branches. `9241_pegase` has a total load of 300,000 MW (others have a load around 50,000 MW), which explains why the prediction error seems larger: in percentage terms, the accuracy is similar.

D. Performance of the Learning Models

This section compares the model $\mathbb{O}$ that directly approximates the mapping $\mathcal{O}$ (Equation 7) with the proposed two-stage approach $\mathbb{D}$. The results show that $\mathbb{D}$ outperforms $\mathbb{O}$ and is more scalable. The comparison is performed on the smaller systems, `France_EHV` and `France_LYON`, which

TABLE II: The Power System Networks with Regional Information

Benchmark	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{G}\rvert$	K	$\lvert\mathcal{N}^k\rvert_{k=1}^K$	$\lvert\mathcal{E}^{\leftrightarrow}\rvert$	Nom. Load
France_EHV	1737	2350	1731	290	12	[338, 280, 233, 179, 143, 126, 124, 72, 67, 64, 57, 54]	148	51949 MW
France_Lyon	3411	4499	3273	771	12	[1158, 357, 294, 288, 264, 255, 231, 197, 184, 67, 62, 54]	219	52394 MW
4661_sdet	4661	5997	2683	724	22	[921, 724, 661, 352, 306, 197, ..., 73, 69, 66, 61, 58, 50]	269	88204 MW
France	6705	8962	6262	1708	12	[1156, 796, 748, 746, 627, 517, 497, 395, 325, 322, 298, 278]	326	54708 MW
9241_pegase	9241	16049	4895	1445	24	[1354, 1203, 1078, 985, 809, 682, ..., 59, 54, 52, 33, 22, 13]	402	312354 MW

	$\hat{p}^f$ (MW)		v (P.U)	
Benchmark	Avg	95% Quantile	Avg	95% Quantile
France_EHV	3.43	11.91	$25 \cdot 10^{-5}$	$76 \cdot 10^{-5}$
France_Lyon	1.25	4.89	$27 \cdot 10^{-5}$	$82 \cdot 10^{-5}$
4661_sdet	2.15	7.86	$117 \cdot 10^{-5}$	$305 \cdot 10^{-5}$
France	0.99	4.11	$50 \cdot 10^{-5}$	$153 \cdot 10^{-5}$
9241_pegase	9.96	35.59	$70 \cdot 10^{-5}$	$222 \cdot 10^{-5}$

TABLE III: Absolute Errors for the Voltage Magnitude at the Coupling Buses and the Active Power Flow of the Coupling Branches.

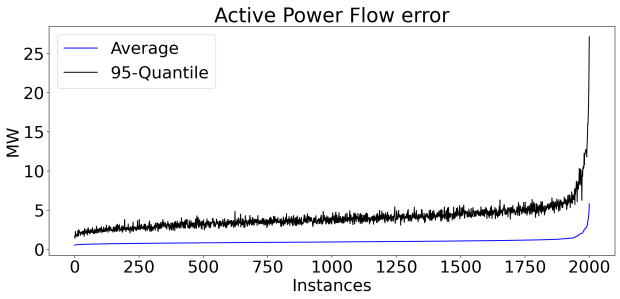


Fig. 6: Prediction Errors (Average and 95% Quantile) over all Testing Instances for the Active Flow of the Coupling Branches for Testcase France. The Instances are Sorted in Increasing Order of Average Error.

represent the high-voltage French system and the high-voltage French with a detailed representation of the Lyon region. Experimental results on the full French system are only given for model $\mathbb{D}$, since the original model exceeds the capacity of the GPU memory. Results for model $\mathbb{D}$ are also reported for two additional benchmarks from the pglib library [34], 4661_sdet and 9241_pegase which include zonal information. The comparison consists of three parts. The first part reports the accuracy for variables $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{p}}^g)$ (that are directly predicted) and the indirectly predicted $\hat{\boldsymbol{p}}^f$. The second part considers constraint violations. The third part discusses how the predictions can be used to seed an optimization model that restores feasibility.

1) Prediction Accuracy: Figure 7 illustrate the convergence of the two models for the predicted variables $\hat{\boldsymbol{v}}, \hat{\boldsymbol{p}}^g$ for a specific bus and generator from the France_LYON test case. The x-axis corresponds to test instances sorted by increasing system load. There is significant volatility in the ground truth values since instances that are close in the x-axis do not necessarily correspond to similar load vectors. Indeed, a similar overall system load may exhibit significant regional load differences. The results demonstrate that, for voltage magnitudes, model $\mathbb{O}$ has significant errors. The same hold for active power. In constrast, model $\mathbb{D}$ closely follows the ground truth for voltage magnitudes and exhibits minor errors

	Model $\mathbb{O}$		Model $\mathbb{D}$	
Benchmark	Avg	95% Quantile	Avg	95% Quantile
France_EHV	$39 \cdot 10^{-5}$	$125 \cdot 10^{-5}$	$22 \cdot 10^{-5}$	$61 \cdot 10^{-5}$
France_Lyon	$45 \cdot 10^{-5}$	$127 \cdot 10^{-5}$	$22 \cdot 10^{-5}$	$78 \cdot 10^{-5}$
4661_sdet	-	-	$78 \cdot 10^{-5}$	$258 \cdot 10^{-5}$
France	-	-	$25 \cdot 10^{-5}$	$84 \cdot 10^{-5}$
9241_pegase	-	-	$39 \cdot 10^{-5}$	$126 \cdot 10^{-5}$

TABLE IV: Prediction Errors for Voltage Magnitudes.

	Model $\mathbb{O}$		Model $\mathbb{D}$	
Benchmark	Avg	95% Quantile	Avg	95% Quantile
France_EHV	8.41	50.82	0.84	3.27
France_Lyon	8.93	47.54	0.30	0.94
4661_sdet	-	-	2.45	12.27
France	-	-	0.19	0.70
9241_pegase	-	-	3.31	16.52

TABLE V: Prediction Errors (MW) for Active Power.

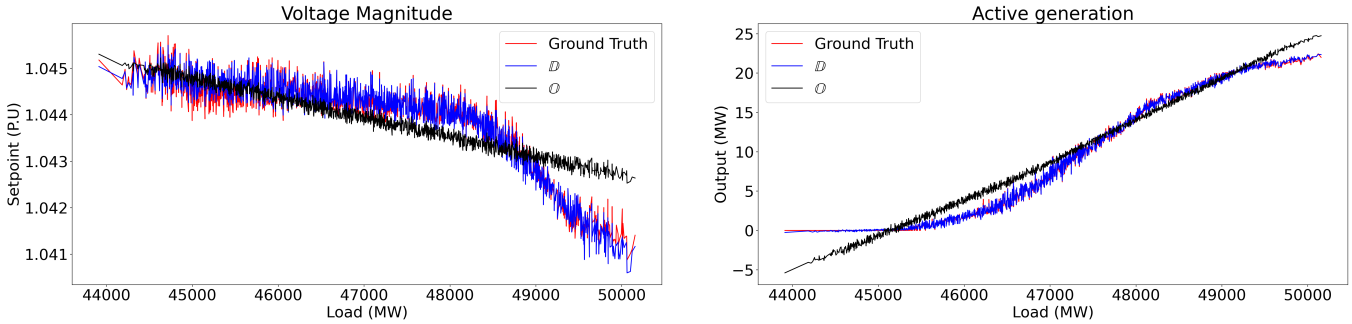
	Model $\mathbb{O}$		Model $\mathbb{D}$	
Benchmark	Avg	95% Quantile	Avg	95% Quantile
France_EHV	4.53	16.88	2.01	4.20
France_Lyon	8.43	32.20	0.82	1.91
4661_sdet	-	-	1.38	4.48
France	-	-	0.45	1.04
9241_pegase	-	-	0.91	3.95

TABLE VI: Prediction Errors (MW) for Active Power Flows.

	Model $\mathbb{O}$		Model $\mathbb{D}$	
Benchmark	v	p^g	v	p^g
France_EHV	99.90	98.46	99.74	99.94
France_Lyon	99.72	99.24	99.91	99.99
4661_sdet	-	-	99.78	99.79
France	-	-	99.97	99.99
9241_pegase	-	-	99.50	99.50

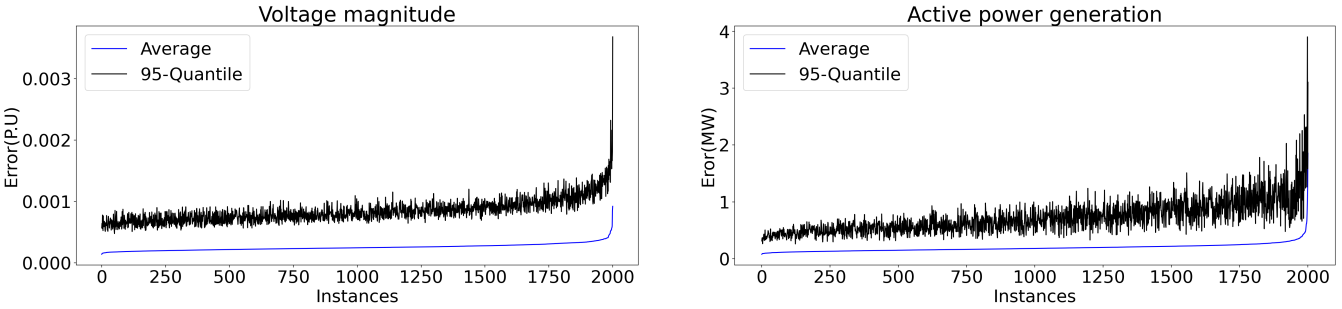
TABLE VII: Percentage of AC-OPF bound Constraints with Violations under 1 MW (for $\hat{p}^g$) and under 10^{-4} P.U. (for $\hat{v}$)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPWRS.2021.3124726, IEEE Transactions on Power Systems



(a) Voltage Magnitude (b) Active power

Fig. 7: Convergence of $\mathbb{O}$ and $\mathbb{D}$ Illustrated for a Bus and Generator.



(a) Voltage Magnitude (b) Active power

Fig. 8: Prediction Errors for the `France` System using Model $\mathbb{D}$.

	Model $\mathbb{O}$		Model $\mathbb{D}$	
Benchmark	Avg	95% Quantile	Avg	95% Quantile
`France_EHV`	4.49	16.20	4.82	9.77
`France_Lyon`	18.65	100.36	1.91	4.67
`4661_sdet`	-	-	2.56	9.76
`France`	-	-	1.05	2.39
`9241_pegase`	-	-	1.04	3.41

TABLE VIII: Violations of Active Power Balance Constraints (MW).

	PROJ($\mathbb{O}$)		PROJ($\mathbb{D}$)		PF($\mathbb{D}$)	
Benchmark	Avg	Max	Avg	Max	Avg	Max
`France_EHV`	0.036	0.193	0.026	0.119	0.012	0.053
`France_Lyon`	0.281	0.987	0.016	0.071	0.002	0.022
`4661_sdet`	-	-	0.058	0.093	0.013	0.031
`France`	-	-	0.012	0.030	0.006	0.023
`9241_pegase`	-	-	0.029	0.059	0.014	0.032

TABLE IX: Differences in Objective Values (in %) between the Feasibility Restorations and the AC-OPF Objective.

Benchmark	Model PROJ($\mathbb{O}$)	Model PROJ($\mathbb{D}$)
`France_EHV`	23	17
`France_Lyon`	35	13
`4661_sdet`	–	72
`France`	–	146
`9241_pegase`	–	172

TABLE X: Average l_1 Distances (in MW) Between the Projection Restorations and the AC Dispatch.

for active power predictions. The difference between the two models is quite striking.

Tables IV, V, and VI summarize the prediction errors over all testcases, buses, generators, and lines, as well the 95% Quantile. The tables omit all power results for generators that are either off for all instances (due to potentially high cost) or constantly producing at their respective upper bounds (due to low cost). For voltage magnitudes, model $\mathbb{D}$ divides the error in half compared to model $\mathbb{O}$. This difference is significant for the prediction of the power flows and constraint violations. Figure 8 demonstrates that model $\mathbb{D}$ scales to the size of the `France` system and continues to produce highly accurate predictions. For active power, model delivers predictions whose errors are an order of magnitude smaller than those of model $\mathbb{O}$. The average errors are below 1 MW, which is small compared to the total system load ($\sim 50,000$ MW). Again, Figure 8 demonstrates that model $\mathbb{D}$ nicely scales to the `France` system. The benefits of model $\mathbb{D}$ are abundantly clear for the power flow predictions $\hat{p}^f$, which are indirectly predicted as a function of the predictions $\hat{v}$ and $\hat{\theta}$. For `France_LYON`, the second largest test case, model $\mathbb{O}$ results in large errors (up to 50 MW). In contrast, model $\mathbb{D}$ results in minor errors, with 95% of the predictions having an error of at most 1.04 MW

	(2)		(3r)		(3i)		(4)	
Benchmark	Sat (%)	Avg (P.U)	Sat (%)	Avg (MW)	Sat (%)	Avg (MVaR)	Sat (%)	Avg (MVA)
France_EHV	99.47	1.6×10^{-4}	99.69	2.98	99.57	0.79	100.00	–
France_Lyon	99.77	1.2×10^{-4}	99.91	1.88	98.78	0.12	100.00	–
4661_sdet	99.88	2.9×10^{-4}	99.83	2.90	99.07	3.92	99.86	0.85
France	99.88	2.0×10^{-4}	99.95	0.94	99.78	0.13	99.99	1.10
9241_pegase	99.49	2.3×10^{-4}	99.97	5.65	98.86	1.43	99.98	3.80

TABLE XI: Percentage of Satisfied Constraints and Average Violations for Violated Constraints of PF($\mathbb{D}$).

	Proj($\mathbb{O}$)		Proj($\mathbb{D}$)		PF($\mathbb{D}$)		AC-OPF	
Benchmark	Avg	Max	Avg	Max	Avg	Max	Avg	Max
France_EHV	2.98	4.49	2.90	4.04	0.57	0.75	4.42	5.31
France_Lyon	6.41	11.35	6.33	9.30	1.19	1.46	9.10	14.47
4661_sdet	-	-	11.53	14.17	2.18	2.31	13.67	15.25
France	-	-	24.14	30.82	3.02	3.46	35.50	44.61
9241_pegase	-	-	31.66	41.85	3.97	4.91	36.86	59.26

TABLE XII: Computing Times of the Feasibility Restorations and AC-OPF (in Seconds).

in the largest benchmark. Compared to the overall system scale, these errors are small in percentage. Note that accurate predictions for power flows are critical for low violation degrees of the AC-OPF constraints.

2) Feasibility: Table VII reports the constraint violations for the bounds on active power and voltage magnitude (constraints (2), (3r)). Model $\mathbb{D}$ has minor violations for 99.9% of the active balance constraints (Table VIII). Again, model $\mathbb{D}$ has an average violations of 1.05 MW in the France test case, which is insignificant compared to the scale of the system.

3) Feasibility Restoration Analysis: This section shows how to use model $\mathbb{D}$ for applications requiring a high-quality *feasible* solution. Table IX reports the differences in objective values between the feasibility restoration and the AC-OPF solution, i.e.,

$$\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |1 - \frac{cost}{cost_{AC}}| \times 100\%$$

where $cost$ denotes the cost after restoration and $cost_{AC}$ denotes the AC-OPF cost. Proj($\mathbb{D}$) is within $< 0.06\%$ on average of the AC-OPF solution and one magnitude smaller compared to Proj($\mathbb{O}$) on France_Lyon. Moreover, the objective value difference is on average $< 0.015\%$ for all benchmarks when the predictions from $\mathbb{D}$ seed the power-flow restoration. The power-flow restoration always satisfies the physical constraints but may have some minor violations of bound constraints. Table XI reports these violations and shows that at least 98.5%, and typically 99.5%, of the bound constraints are satisfied and that the violations are minor. More advanced power flows would further reduce these minor violations. Note that these violations explain why the objective values of Proj($\mathbb{D}$) may be higher than those of PF($\mathbb{D}$).

In terms of computational efficiency, model $\mathbb{D}$ delivers a prediction in a few milliseconds, which makes it sufficient to compare the optimization results only. Table XII compares the execution times of the feasibility restoration and the AC-OPF optimizations. The results demonstrate that the power-flow optimization is significantly faster compared to the AC-OPF optimization for all the benchmarks. Its running time is under 5 seconds even in the largest testcase. *This indicates that a combination of machine learning and optimization is beneficial to speed-up AC-OPF optimization with neglibigle objective difference and operational bound violations.*

VII. Discussion

There is a complexity trade-off between the two stages of the approach. The size of the first-stage model depends on the number of coupling branches ($\mathcal{E}^{\leftrightarrow}$), while the complexity of the second stage depends on the maximum region size ($\max_{k\in[K]} |\mathcal{N}^k|$). Decomposing the system into fewer number of regions leads to smaller $|\mathcal{E}^{\leftrightarrow}|$ and a smaller learning model for the first stage, but results in larger second-stage models which might be similar in size to the entire system, thus negating any benefits of the two-stage approach. Conversely, decomposing the system into a larger number of regions leads to smaller second-stage models, but also a larger learning model for the first stage.

In per-unit representation, small errors in voltage magnitudes may translate into significant error in AC power flows, due to the physics of power-flow equations (5r)-(5i) which involve a series of multiplications between voltage and admittance values. Branches with low impedance values (e.g., shorter lines) will result in large admittance values, hence boosting the errors on the power flows due to voltage prediction errors. It is an interesting research direction to investigate how best to decouple a network for machine learning. Intuitively, the first-stage learning should differentiate branches based on admittance values, with more focus on selecting coupling branches with higher impedance, thus ensuring that minor inaccuracies in voltage magnitude will not lead to large errors for the indirectly predicted power flows.

There are several interesting avenues to broaden the scope of the proposed approach, First, incorporating line and generator contingencies is critical in actual operations. Appendix IX-B

presents a transfer learning approach for $N-1$ line contingencies and preliminary results are encouraging. Second, actual operations rely on a set of commitment decisions, generator bids, and renewable generation predictions; these vary over time and thus raise interesting challenges. In this context, the fast training procedure proposed in this paper has a significant advantage over slow training schemes since it can be trained after the Day-ahead Unit Commitment where the generator bids and commitments for the next day are known. A thorough comparison of all these results is needed however, to compare different approaches.

VIII. Conclusion

This paper considered the design of a fast and scalable training for a machine-learning model that predicts AC-OPF solutions. It was motivated by the facts that (1) more accurate forecasts, topology optimization, and the stochasticity induced in renewable energy may lead to fundamentally different AC-OPF instances; and (2) existing machine-learning algorithms for AC-OPF require significant training time and do not scale to the size of real transmission systems. The paper proposed a novel 2-stage approach that exploits a spatial network decomposition. The power network is viewed as a set of regions, the first stage learns to predict the flows and voltages on the coupling buses and lines, and the second stage trains, in parallel, the machine-learning models for each region. Experimental results on the French transmission system (up to 6,700 buses) and `pglib` test cases with up to 9,000 buses) demonstrate the potential of the approach. Within a training time of 90 minutes, the approach predicts AC-OPF solutions with very high fidelity (e.g., an average error of 1 MW for an overall load of 50 GW) and minor constraint violations, producing significant improvements over the state-of-the-art. The predictions can then seed a power flow to eliminate the physical constraint violations, resulting in minor violations only for the operational bound constraints. Future work will focus on generalizing the approach to security-constrained OPF, by studying how to merge the algorithm proposed in [3] to the AC setting and the proposed 2-stage approach, and applying the method to reinforcement-learning approaches to real-time optimal power flows.

References

[1] K. Lehmann, A. Grastien, and P. Van Hentenryck, "Ac-feasibility on tree networks is np-hard," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 798–801, 2016.

[2] A. Monticelli, M. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Transactions on Power Systems*, vol. 2, no. 1, pp. 175–180, 1987.

[3] A. Velloso, P. Van Hentenryck, and E. S. Johnson, "An exact and scalable problem decomposition for security-constrained optimal power flow," *Electric Power Systems Research*, vol. 195, p. 106677, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779620304806

[4] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1319–1327, 2008.

[5] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1346–1355, Aug 2008.

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

[7] Niharika, S. Verma, and V. Mukherjee, "Transmission expansion planning: A review," in *International Conference on Energy Efficient Technologies for Sustainability*, April 2016, pp. 350–355.

[8] M. Chatzos, F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, "High-fidelity machine learning approximations of large-scale optimal power flow," 2020. [Online]. Available: https://arxiv.org/pdf/2006.16356

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

[10] L. Duchesne, E. Karangelos, and L. Wehenkel, "Machine learning of real-time power systems reliability management response," in *2017 IEEE Manchester PowerTech*, 2017, pp. 1–6.

[11] F. Hasan, A. Kargarian, and A. Mohammadi, "A survey on applications of machine learning for optimal power flow," in *2020 IEEE Texas Power and Energy Conference (TPEC)*, 2020, pp. 1–6.

[12] S. Misra, L. Roald, and Y. Ng, "Learning for constrained optimization: Identifying optimal active constraint sets," 2019. [Online]. Available: https://arxiv.org/pdf/1802.09639

[13] A. S. Xavier, F. Qiu, and S. Ahmed, "Learning to solve large-scale security-constrained unit commitment problems," *INFORMS Journal on Computing*.

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

[15] F. Hasan, A. Kargarian, and J. Mohammadi, "Hybrid learning aided inactive constraints filtering algorithm to enhance ac opf solution time," *IEEE Transactions on Industry Applications*, vol. 57, no. 2, pp. 1325–1334, 2021.

[16] A. Robson, M. Jamei, C. Ududec, and L. Mones, "Learning an optimally reduced formulation of opf through meta-optimization," 2020. [Online]. Available: https://arxiv.org/abs/1911.06784

[17] K. Baker, "A learning-boosted quasi-newton method for ac optimal power flow," 2020. [Online]. Available: https://arxiv.org/abs/2007.06074

[18] K.Baker, "Learning warm start points for ac optimal power flow," 2019. [Online]. Available: https://arxiv.org/pdf/1905.08860

[19] L. Chen and J. E. Tate, "Hot-starting the ac power flow with convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/pdf/2004.09342

[20] X. Pan, T. Zhao, and M. Chen, "Deepopf: Deep neural network for dc optimal power flow," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, pp. 1–6.

[21] X. Pan, M. Chen, T. Zhao, and S. H. Low, "Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems," 2020. [Online]. Available: https://arxiv.org/abs/2007.01002

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

[23] A. Venzke and S. Chatzivasileiadis, Verification of neural network behaviour: Formal guarantees for power system applications, *IEEE Transactions on Smart Grid*, 2020.

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

[25] X. Pan, T. Zhao, M. Chen, and S. Zhang, "Deepopf: A deep neural network approach for security-constrained dc optimal power flow," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1725–1735, 2021.

[26] A. Velloso and P. Van Hentenryck, "Combining deep learning and optimization for preventive security-constrained dc optimal power flow," *IEEE Transactions on Power Systems*, pp. 1–1, 2021.

[27] Y. Zhou, B. Zhang, C. Xu, T. Lan, R. Diao, D. Shi, Z. Wang, and W.-J. Lee, "A data-driven method for fast ac optimal power flow solutions via deep reinforcement learning," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1128–1139, 2020.

[28] Z. Yan and Y. Xu, "Real-time optimal power flow: A lagrangian based deep reinforcement learning approach," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3270–3273, 2020.

[29] J. H. Woo, L. Wu, J.-B. Park, and J. H. Roh, "Real-time optimal power flow using twin delayed deep deterministic policy gradient algorithm," *IEEE Access*, vol. 8, pp. 213 611–213 618, 2020.

[30] E. R. Sanseverino, M. L. Di Silvestre, L. Mineo, S. Favuzza, N. Q. Nguyen, and Q. T. T. Tran, "A multi-agent system reinforcement learning based optimal power flow for islanded microgrids," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016, pp. 1–6.
[31] J. D. D. Glover and M. S. Sarma, *Power System Analysis and Design*, 3rd ed. USA: Brooks/Cole Publishing Co., 2001.
[32] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.
[33] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[34] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, C. Josz, R. Korab, B. Lesieutre, J. Maeght, T. W. K. Mak, D. K. Molzahn, T. J. Overbye, P. Panciatici, B. Park, J. Snodgrass, A. Tbaileh, P. V. Hentenryck, and R. Zimmerman, "The power grid library for benchmarking ac optimal power flow algorithms," 2021.

IX. Appendix

A. Sensitivity to the Size of the Training Set

This section studies the sensitivity of the first-stage model $\mathbb{D}$, which learns the mapping $\mathcal{O}^{\leftrightarrow}$, to the size of the training set. It considers the two largest test cases, `France` and `9241_pegase`, and training sets that consist of 1, 10, 50, 100 percents of the original training set (8,000 entries). Figure 9 reports the accuracy as a function of the size of the training set for `9241_pegase`: the testing error significantly decreases when the size of the training set increases from 1% to 10%. Moreover, there is a monotone decrease in the error, both in the average and for the 95% quantile, when the training size increases from 10% to 100%. The behavior is similar for `France` (Figure 10). Note that the errors for the 100% entries correspond to the values reported in Table III for the coupling flows.

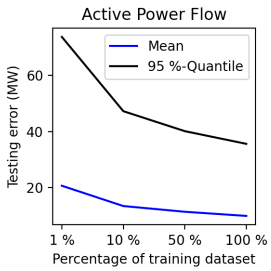


Fig. 9: `9241_pegase`

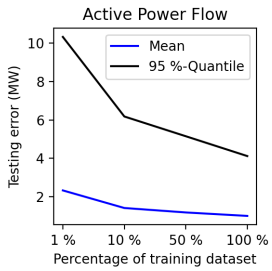


Fig. 10: `France`

B. Transfer Learning under Line Contingency

This section presents preliminary results about a very fast procedure to re-train a learning model when a line contingency occurs. The case study can be summarized as follows.

1) The operator has trained a collection of (first and second-stage) models that approximate the behavior of the system in the nominal case.
2) A line contingency occurs in the system and the operator would like a machine-learning model to approximate the behavior of the power system under this contingency.
3) Training data for this contingency is available from reliability studies and high-fidelity simulations.

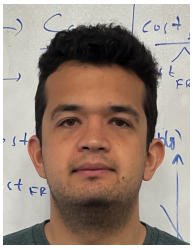
This section explores how to exploit the weights of existing machine-learning models for the nominal case in order to generate new machine-learning models for the contingency state. For simplicity, it focuses on contingencies for the coupling lines, which are particularly important. Indeed, the experiments consider lines with some of the largest flows: for `France` and `9241_pegase`, these lines carry $\sim$ 500 and $\sim$ 1500 MW, respectively. Like in the nominal case, the experiments assume a training set of $8,000$ entries. The training is given 15 minutes which is the frequency of the Look Ahead Commitment (LAC) for an ISO like MISO. The first stage is allocated 5 minutes and the second stage is allocated the remaining time. The results compare two training methods:

- Cold: The cold-start training does not consider a-priori information from the nominal case.
- Warm: The warm-start training uses the weights of the nominal case as initial values for the contingency state.

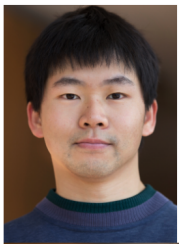
Comprehensive results are reported in Table XIII for the largest two benchmarks, `France` and `9241_pegase`, where five line contingencies are considered for each benchmark. For comparison purposes, the results for the cold-start (90 minutes) training on the nominal case are displayed again as the first line of each test case. The main take-away is that the warm-start training (15 minutes) for a contingency has prediction errors and constraints violations similar to the cold-start training (90 minutes) for the nominal case, which demonstrates the possibility of very fast training. The warm-start training significantly outperforms the cold-start training for these contingencies, indicating a strong correlation in the OPF behavior between the nominal case and the contingency cases. In terms of active power p^g, the warm-start model halves the error of the cold-start model for `France` and produces noticeable improvements for `9241_pegase`. The error differences are significant for voltage magnitudes and active power flows, with improvements by one order of magnitude. Finally, the warm-start training benefits are evident for constraint violations with one order of magnitude improvement for the power balance constraint ($6r$) in both benchmarks. Regarding the operational bound constraints (2) and ($3r$), the improvement is also clear especially in the `9241_pegase` case. Overall, these results highlight a promising avenue for the decomposition methodology proposed in the paper: its ability to train machine-learning models very fast in response to contingencies.

	p^g error (MW)		v error (P.U. $\times 10^{-5}$)		p^f error (MW)		(6r) viol. (MW)		(2) sat. (%)		(3r) sat. (%)	
Benchmark	Warm	Cold	Warm	Cold	Warm	Cold	Warm	Cold	Warm	Cold	Warm	Cold
France - nom	–	0.19	–	25	–	0.45	–	1.05	–	99.97	–	99.99
France - 1	0.24	0.81	20	265	1.00	22.53	2.49	57.78	99.97	99.86	99.99	99.75
France - 2	0.27	0.53	19	152	1.02	15.10	2.51	38.63	99.78	99.68	99.98	99.90
France - 3	0.22	0.55	19	193	0.80	19.53	1.97	50.24	99.97	99.58	99.99	99.82
France - 4	0.27	0.55	19	188	0.85	17.42	2.06	44.67	99.97	99.64	99.98	99.82
France - 5	0.21	0.56	18	188	0.82	14.76	2.04	37.52	99.97	99.80	99.98	99.89
9241_pegase - nom	–	3.31	–	39	–	0.91	–	1.04	–	99.50	–	99.50
9241_pegase - 1	3.80	6.80	39	204	1.05	10.11	1.43	28.71	99.31	98.81	99.14	97.31
9241_pegase - 2	4.82	5.46	40	165	1.11	5.85	1.66	15.27	99.74	98.32	98.86	97.77
9241_pegase - 3	3.46	4.54	34	116	0.99	6.49	1.29	17.49	99.63	98.83	99.34	97.84
9241_pegase - 4	4.51	5.32	37	131	1.11	6.22	1.58	16.58	99.53	99.01	98.90	97.07
9241_pegase - 5	3.50	4.89	35	164	0.96	5.98	1.16	15.73	99.61	97.54	99.42	97.89

TABLE XIII: Accuracy of Warm & Cold-Start Models For Contingency Case.



Minas Chatzos received the M.Eng. degree in Electrical & Computer Engineering from the National Technical University of Athens, Athens, Greece and is currently working towards a Doctoral Degree in Operations Research at the Georgia Institute of Technology, Atlanta, GA. His research interests includes Machine Learning and Optimization under uncertainty with applications in Energy.



Terrence W.K. Mak is a Postdoctoral Fellow at School of Industrial Systems & Engineering, Georgia Tech. He is an interdisciplinary researcher with working on the intersections between Optimization, Data Analytics, and Machine Learning, with broad applications to industrial Power & Energy Systems. He has published more than 18 peer-reviewed papers in various academic domains, and have experience in collaborations with various national labs, industrial utilities, and universities.

Pascal Van Hentenryck Pascal Van Hentenryck is the A. Russell Chandler III Chair and Professor in the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology and the Associate Chair for Innovation and Entrepreneurship. He is also the director of the NSF AI for Advances in Optimization. Van Hentenryck is an INFORMS Fellow and a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI), the recipient of two honorary doctoral degrees, and teaching excellence awards at Brown University and Georgia Tech. Several of his optimization systems have been in commercial use for more than 20 years. His current research focuses on machine learning, optimization, and privacy with applications in energy and transportation.