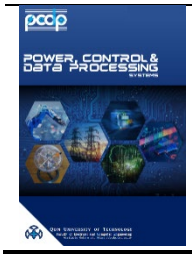
DOI: 10.30511/pcdp.2026.2096254.1104 - ISSN: 3060-8376 - Vol 4(1), 2027



POWER, CONTROL AND DATA PROCESSING SYSTEMS

Available Online at: https://pcdp.qut.ac.ir/

Deep Operator Learning-Based Prediction of Primal and Dual Variables in AC Optimal Power Flow

ARTICLE INFO

Article Type
Original Research

Authors
Reza Mohammad Yousefi[1]*
Ahmad Salehi Dobakhshari[1]

[1] Faculty of Engineering, University of Guilan, Guilan, Iran

*** Correspondence***
Address: Faculty of Engineering, University of Guilan, Guilan, Iran

reza_yousefi@webmail.guilan.ac.ir
salehi_ahmad@guilan.ac.ir

Article History
Received: July 31, 2026
Accepted: September 24, 2026
ePublished: March 01, 2027

ABSTRACT

This paper proposes a deep operator learning framework for fast and accurate prediction of both primal and dual variables in alternating current optimal power flow (AC-OPF). Unlike conventional neural network surrogates that learn a fixed input–output mapping and often ignore dual information, the proposed method is designed to approximate the underlying operator that maps varying system operating conditions to optimal OPF solutions. The framework is built on a primal–dual DeepONet architecture, where branch and trunk networks jointly encode system conditions and query locations to predict generator dispatch, voltage states, and associated Lagrange multipliers. To better capture optimization structure, a differentiable dual completion layer is incorporated to improve dual feasibility and complementary slackness, and a KKT-informed loss function is used to guide training toward physically consistent and optimization-aware solutions. The proposed model is evaluated on standard IEEE benchmark systems, where it achieves high prediction accuracy for both primal and dual variables, millisecond-level inference time, and significantly reduced KKT residuals compared with conventional learning-based OPF approximations. These results highlight the promise of deep operator learning as an efficient surrogate for real-time AC-OPF analysis and power-system operation.

Keywords: Deep Operator Learning; DeepONet; AC Optimal Power Flow; Primal–Dual Learning; Lagrange Multipliers; KKT Conditions; Power System Optimization.

1 Introduction

Optimal power flow (OPF) is a core problem in power-system operation and planning. In its alternating-current form, AC-OPF determines an economically efficient operating point while enforcing the nonlinear AC power-flow equations together with generator capability limits, voltage bounds, reactive-power limits, and transmission constraints [1]. Because AC-OPF is inherently nonlinear and nonconvex, solving it repeatedly for large-scale systems remains computationally intensive, especially in settings that demand fast or real-time decision making [2].

This challenge has become more pronounced with the increasing penetration of renewable generation, distributed energy resources, flexible demand, and other sources of operational uncertainty. In modern grid operation, applications such as real-time energy management, market clearing, contingency screening, and security assessment require the solution of large numbers of OPF instances under continuously changing operating conditions. As a result, there is growing interest in learning-based surrogate models that can approximate AC-OPF solutions with substantially lower online computational cost while preserving acceptable accuracy and feasibility [3].

Existing learning-based AC-OPF methods may be broadly grouped into four categories. The first category comprises supervised regression methods that learn a direct mapping from operating conditions to optimal solutions using standard neural networks. These approaches typically provide very fast inference, but they treat AC-OPF as a fixed-dimensional vector regression task and therefore do not explicitly exploit its underlying functional structure [4]. The second category includes physics-informed methods, which incorporate power-flow equations and operational constraints into the training objective in order to improve physical consistency. The third category consists of primal–dual learning methods that jointly predict decision variables and Lagrange multipliers, thereby capturing not only optimal operating points but also sensitivity and congestion information. The fourth category, emerging from scientific machine learning, is operator learning, which seeks to approximate mappings between function spaces rather than between finite-dimensional vectors [5, 6, 7, 8]. Although these lines of research have each advanced the state of learning-based OPF, there remains limited work that unifies operator learning, primal–dual prediction, and refinement within a single framework [9].

From an operator-learning perspective, AC-OPF can be interpreted as a nonlinear mapping from distributed operating conditions to optimal primal and dual quantities. The inputs include active and reactive power injections across the network, while the outputs comprise voltage states, generator dispatch, branch flows, and Lagrange multipliers. Because these quantities are naturally associated with buses, generators, and transmission branches, the operator-learning formulation preserves the physical and spatial structure of the power network rather than treating the AC-OPF problem as an unstructured vector-to-vector mapping.

Within this perspective, Deep Operator Networks (DeepONets) are particularly well suited to AC-OPF. A DeepONet approximates an operator through two coordinated components: a branch network that encodes the global input function and a trunk network that evaluates the output at specified query locations. In power systems, these query locations correspond naturally to buses, generators, and transmission lines. This structure offers several advantages. First, it preserves the network-indexed nature of AC-OPF inputs and outputs, rather than collapsing them into a single fixed vector. Second, it supports the learning of solution functions evaluated at different system entities, which aligns well with the spatially distributed character of voltages, flows, and dual variables. Third, it provides a principled mechanism for approximating the underlying nonlinear operator itself, rather than only fitting a discretization-specific input–output map. For these reasons, DeepONet offers a strong architectural match to the functional structure of AC-OPF [5, 8].

The case for predicting dual variables is equally compelling. In power-system operation, dual variables carry important economic and operational meaning: they quantify marginal costs, reveal congestion patterns, indicate binding constraints, and support sensitivity analysis. However, most learning-based OPF studies focus primarily on primal variables. In principle, dual variables can be recovered from an exact primal solution through the Karush–Kuhn–Tucker (KKT) conditions, but in practice this recovery is highly sensitive to active-set identification and to errors in the approximate primal solution [10]. Even small inaccuracies in predicted primal variables can lead to incorrect active constraints and large multiplier errors [11]. Moreover, dual variables are often more difficult to learn directly because they may change sharply near active-set transitions and must satisfy nonnegativity, stationarity, and complementarity conditions simultaneously [12]. These observations suggest that accurate primal–dual prediction requires both an expressive operator-learning architecture and explicit mechanisms for enforcing optimization structure.

Motivated by these considerations, this paper casts AC-OPF as a primal–dual operator-learning problem. Instead of learning only a fixed vector-to-vector surrogate, the proposed framework learns the nonlinear operator that maps distributed operating conditions to distributed optimal primal and dual solutions. To realize this idea, we develop a DeepONet-based model equipped with a differentiable dual-completion layer and a KKT-informed training objective. Together, these components are designed to improve physical feasibility, complementarity, and overall optimization consistency.

The main contributions of this paper are summarized as follows:

- A primal–dual DeepONet architecture is developed to jointly predict generator dispatch, voltage states, branch quantities, and the associated Lagrange multipliers within a unified operator-learning framework.

- A differentiable dual-completion layer is introduced to promote essential structural properties of the learned multipliers, including nonnegativity and complementarity, thereby mitigating inconsistencies commonly associated with direct dual regression.

- A KKT-informed operator loss is proposed to embed the first-order optimality conditions of AC-OPF into training, improving physical feasibility and optimization consistency compared with purely supervised learning.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive review of the existing literature regarding learning-based OPF surrogates, physics-informed methodologies, and deep operator learning. In Section 3, the AC-OPF problem is formally cast as an operator-learning task. Section 4 elucidates the architecture of the proposed primal–dual DeepONet, specifically detailing the differentiable dual-completion layer and the KKT-informed loss function. The training protocols and inference procedures are described in Section 5. Section 6 presents an extensive numerical evaluation, including robustness and feasibility analyses, while Section 7 concludes the paper with a summary of findings and directions for future research.

2 Related Work and Research Gap

2.1 Emerging Machine Learning Paradigms in Modern Power and Complex Systems

In recent years, data-driven and machine learning (ML) methodologies have witnessed widespread adoption across complex engineering domains, operational diagnostics, and energy systems. In modern renewable energy and power networks, ML paradigms have proven essential for time-series forecasting, state estimation, and condition monitoring. For instance, hybrid deep learning architectures combining long short-term memory (LSTM) networks with attention mechanisms have been successfully deployed for volatile wind power forecasting [13], while advanced classification algorithms enable accurate fault detection and identification in photovoltaic (PV) arrays [14]. Concurrently, the exploration of graph topology and spatial representations has significantly enhanced relational inference and link prediction in networked systems [15]. Furthermore, specialized hybrid architectures fusing convolutional, recurrent, and probabilistic deep models (such as CNN-RNN-RBM networks) have demonstrated remarkable efficacy in extracting intricate spatio-temporal dynamics from high-dimensional signal streams [16]. Building upon these foundational data-driven successes, recent research has increasingly turned toward adapting deep learning models to large-scale grid optimization and dispatch challenges, such as the OPF problem.

2.2 Learning-Based Surrogates for Optimal Power Flow

Machine learning has been widely investigated as a tool for accelerating OPF by replacing repeated numerical optimization with fast surrogate models. A large portion of the literature focuses on supervised learning, where models are trained on historical or solver-generated datasets to approximate the mapping from operating conditions, such as active and reactive load levels, to optimal generator dispatch and network states. Representative studies have demonstrated that neural-network-based surrogates can substantially reduce computation time for DC-OPF and AC-OPF while maintaining acceptable accuracy across many operating regimes [1, 17, 18].

Despite these advances, most existing methods formulate OPF approximation as a fixed-dimensional vector-to-vector regression task. In this setting, the input is usually represented by a stacked feature vector and the output by a finite-dimensional solution vector. Although this formulation is computationally convenient, it does not explicitly exploit the fact that both operating conditions and OPF solutions are spatially distributed over the power network. Consequently, standard surrogate models may be sensitive to changes in spatial demand and generation patterns and may generalize poorly when predictions are required over variable query locations or more structured operating scenarios [19].

2.3 Learning-Based and Heuristic Optimization in Modern Grid and Energy Systems

Beyond classical supervised surrogates for centralized OPF, machine learning and advanced optimization techniques have gained substantial traction across diverse power-grid and energy-system operational layers. In distribution-level decision making, reinforcement learning formulations such as dueling deep Q-networks (DDQN) have been leveraged to dynamically reconfigure distribution feeders, effectively mitigating power losses and voltage profile deviations across varying load profiles [20]. Data-driven and spatial machine-learning algorithms have likewise been employed to optimize infrastructure allocation under spatial demand uncertainties, including the optimal siting of electric vehicle charging stations [21].

Simultaneously, heuristic and multi-objective optimization schemes are widely utilized to coordinate complex distributed assets, such as energy hub performance scheduling via improved particle swarm optimization (IPSO) [22] and home energy management under volatile renewable injections and battery storage dynamics [23]. Dynamic asset management and system reliability under high renewable penetration further rely on data-driven digital twins for battery degradation and remaining useful life estimation [24], as well as dynamic coherency analysis to support secure islanding and renewable integration [25]. While these heuristic and specialized learning algorithms effectively solve specific operational subproblems, scaling them to system-wide AC-OPF requires scalable architectures that can guarantee physical law satisfaction and exact mathematical consistency.

2.4 Physics-Informed and KKT-Aware OPF Learning

To improve physical consistency, several studies have augmented supervised learning with penalties derived from power-system constraints. Physics-informed approaches commonly include residual terms associated with AC power-flow equations, voltage limits, generator limits, and

Table 1: Comparison of representative learning-based OPF approaches.

Method category	Primal	Dual	KKT-aware	Operator learning
Standard Learning-Based OPF surrogate [1, 18]	✓	×	×	×
Physics-informed OPF learning [17, 26, 27]	✓	×	Partial	×
Primal–dual OPF learning [32, 33, 34]	✓	✓	Partial	×
Proposed primal–dual DeepONet	✓	✓	✓	✓

branch-flow constraints. These formulations aim to reduce infeasibility even when prediction errors remain nonzero. Closely related methods incorporate optimization structure more explicitly through primal–dual learning, Lagrangian-based formulations, or KKT-informed objectives. In such approaches, dual variables and optimality residuals provide additional information about sensitivities, active constraints, and local optimality conditions [26, 27, 28].

Although physics-informed and KKT-aware models improve upon purely supervised regression, most existing methods are still based on finite-dimensional architectures such as multilayer perceptrons. Therefore, they do not explicitly learn the solution operator that maps distributed operating conditions to distributed OPF outputs. In addition, many methods either omit dual-variable prediction or treat dual variables as secondary regression targets without sufficient architectural mechanisms to enforce nonnegativity, complementarity, and stationarity.

2.5 Deep Operator Learning

Deep operator learning has emerged as an effective paradigm for approximating mappings between function spaces. Instead of learning a single mapping between discretized vectors, operator-learning methods seek to approximate the underlying operator that maps an input function to an output function. DeepONet is a prominent architecture in this area. It combines a branch network, which encodes observations of the input function, with a trunk network, which encodes the query location at which the output is evaluated [5, 29].

This perspective is highly relevant to OPF. Spatially distributed active and reactive loads, renewable injections, voltage magnitudes, branch flows, and generator outputs can all be interpreted as functions indexed over buses, generators, or branches. Therefore, an operator-learning framework is better aligned with the structure of OPF than conventional black-box vector regression. However, despite this natural connection, operator-learning methods remain underexplored in the OPF literature, particularly for joint primal–dual prediction.

2.6 Dual Variables and Sensitivity Information in OPF

Dual variables are central to optimization-based power-system analysis. In AC-OPF, multipliers associated with equality constraints encode marginal system sensitivities, whereas inequality multipliers indicate binding limits, congestion patterns, and scarcity conditions. These quantities are important for economic interpretation, active-constraint identification, sensitivity analysis, and decision support [30]. They can also support warm-starting strategies and provide insight into the local structure of optimal operating points [31].

From a machine-learning perspective, however, dual-variable prediction is substantially more challenging than primal-variable prediction. Dual variables may change abruptly near active-set transitions, and naive regression can produce negative multipliers, nonzero multipliers for inactive constraints, or outputs that violate complementary slackness. Therefore, accurate dual prediction requires expressive models as well as structural mechanisms that promote dual feasibility and KKT consistency [32, 33, 34]. Existing OPF surrogates only partially address these requirements, and explicit architectural treatment of dual feasibility and complementarity remains limited.

2.7 Research Gap and Position of This Work

The literature reveals three main gaps. First, most learning-based OPF surrogates treat OPF approximation as finite-dimensional regression rather than operator learning. Second, many prior works focus primarily on primal variables, while dual variables are either ignored or predicted without sufficient structural support. Third, even when optimization-aware losses are used, complementarity and dual feasibility are often encouraged only through penalty terms rather than strengthened through differentiable architectural refinement.

To address these limitations, this paper formulates AC-OPF as a primal–dual operator-learning problem and proposes a DeepONet-based architecture that predicts both primal and dual variables over network locations. A differentiable dual completion layer is introduced to improve nonnegativity, complementarity, and dual consistency, while a KKT-informed loss guides the model toward physically feasible and optimization-consistent predictions. To the best of our knowledge, the combination of operator learning, joint primal–dual prediction, and differentiable KKT-aware dual refinement has not been systematically developed for AC-OPF. Table 1 summarizes the position of the proposed method relative to representative learning-based OPF approaches.

3 AC-OPF as an Operator-Learning Problem

3.1 AC-OPF Formulation

Consider a power network defined by the bus set $\mathcal{N}$, the transmission-branch set $\mathcal{E}$, and the generator-bus set $\mathcal{S} \subseteq \mathcal{N}$. For each bus $i \in \mathcal{N}$, let v_i and θ_i denote the voltage

magnitude and phase angle, respectively. For each generator bus $i \in \mathcal{S}$, let p_i^g and q_i^g denote active and reactive power generation. The active and reactive power demands are denoted by p_i^d and q_i^d. The AC optimal power flow problem is formulated as

$$\min_{\{p_i^g, q_i^g, v_i, \theta_i\}} \quad \sum_{i \in \mathcal{S}} C_i(p_i^g), \tag{1}$$

subject to the generator operating limits

$$\underline{p}_i^g \le p_i^g \le \overline{p}_i^g, \qquad \underline{q}_i^g \le q_i^g \le \overline{q}_i^g, \qquad i \in \mathcal{S}, \tag{2}$$

the voltage-magnitude limits

$$\underline{v}_i \le v_i \le \overline{v}_i, \qquad i \in \mathcal{N}, \tag{3}$$

and the AC branch-flow equations

$$p_{ij}^f = g_{ij} v_i^2 - v_i v_j \left(g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij} \right), (i,j) \in \mathcal{E}, \tag{4}$$

$$q_{ij}^f = -b_{ij} v_i^2 - v_i v_j \left(g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij} \right), (i,j) \in \mathcal{E}, \tag{5}$$

where $\theta_{ij} = \theta_i - \theta_j$, and g_{ij} and b_{ij} are the conductance and susceptance of branch (i,j), respectively.

The active- and reactive-power balance equations are imposed at each bus as

$$p_i^g - p_i^d = \sum_{j:(i,j)\in\mathcal{E}} p_{ij}^f, \qquad i \in \mathcal{N}, \tag{6}$$

$$q_i^g - q_i^d = \sum_{j:(i,j)\in\mathcal{E}} q_{ij}^f, \qquad i \in \mathcal{N}. \tag{7}$$

For buses without generators, p_i^g and q_i^g are set to zero. Transmission-line thermal limits are represented by

$$\left(p_{ij}^f\right)^2 + \left(q_{ij}^f\right)^2 \le \left(\overline{s}_{ij}\right)^2, \qquad (i,j) \in \mathcal{E}, \tag{8}$$

where $\overline{s}_{ij}$ is the apparent-power limit of branch (i,j).

The primal decision vector is defined as

$$x = \{p_i^g, q_i^g, v_i, \theta_i\}_{i \in \mathcal{N}}, \tag{9}$$

where generation variables are active only on $\mathcal{S}$. Let u denote the exogenous operating condition, including load demands and renewable power injections. The AC-OPF problem can then be written compactly as

$$\min_x \ f(x;u) \tag{10}$$

subject to

$$g(x,u) = 0, \tag{11}$$

$$h(x,u) \le 0, \tag{12}$$

where $g(x,u)$ collects the equality constraints, including active- and reactive-power balance equations, and $h(x,u)$ collects the inequality constraints, including generator limits, voltage limits, and line-flow limits.

The associated Lagrangian is

$$\mathcal{L}(x,\lambda,\mu;u) = f(x;u) + \lambda^\top g(x,u) + \mu^\top h(x,u), \tag{13}$$

where λ and μ denote the Lagrange multipliers associated with equality and inequality constraints, respectively. The inequality multipliers satisfy the dual-feasibility condition

$$\mu \ge 0. \tag{14}$$

The first-order optimality conditions are given by

$$\nabla_x \mathcal{L}(x^\star, \lambda^\star, \mu^\star; u) = 0, \tag{15}$$

$$g(x^\star, u) = 0, \tag{16}$$

$$h(x^\star, u) \le 0, \tag{17}$$

$$\mu^\star \ge 0, \tag{18}$$

$$\mu^\star \odot h(x^\star, u) = 0, \tag{19}$$

where $\odot$ denotes elementwise multiplication. These conditions characterize primal feasibility, dual feasibility, stationarity, and complementarity, and they provide the basis for the KKT-informed learning objective developed in this work.

3.2 Operator-Learning Interpretation

Classical learning-based OPF surrogates approximate a finite-dimensional mapping from operating conditions to optimal decision variables. In contrast, this work views AC-OPF as a nonlinear operator that maps a distributed operating condition to distributed primal and dual solution functions. Specifically, the operating condition $u(\cdot)$ is treated as a network-indexed input function, while the optimal primal and dual quantities are treated as output functions defined over buses, generators, and branches.

The corresponding primal–dual solution operator is written as

$$\mathscr{G} : u(\cdot) \mapsto \left(x^\star(\cdot), \lambda^\star(\cdot), \mu^\star(\cdot)\right). \tag{20}$$

Here, $x^\star(\cdot)$ represents the optimal primal solution function, while $\lambda^\star(\cdot)$ and $\mu^\star(\cdot)$ represent the equality- and inequality-dual solution functions, respectively.

This formulation differs from conventional vector regression in two important ways. First, it preserves the network-indexed structure of OPF inputs and outputs instead of treating them as unstructured stacked vectors. Second, it enables the model to learn the underlying solution operator rather than only a discretization-specific input-output mapping. This is particularly useful for AC-OPF because loads, renewable injections, voltage states, generator outputs, branch flows, and Lagrange multipliers are all naturally associated with physical network entities.

Accordingly, the proposed framework learns the operator $\mathscr{G}$ using a DeepONet-based architecture. The branch network encodes the global operating condition $u(\cdot)$, while the trunk network evaluates the learned primal–dual solution functions at specified network entities. The resulting model produces both primal and dual predictions and is trained using supervised solution data together with KKT-informed residual terms that promote feasibility, stationarity, dual feasibility, and complementarity.

3.3 Input and Output Function Representation

Let ξ denote a network location or query index, such as a bus, generator, or branch identifier. The operating condition is represented as an input function

$$u(\xi) = \left[P_D(\xi), Q_D(\xi), s^{\text{hist}}(\xi)\right], \tag{21}$$

where $P_D(\xi)$ and $Q_D(\xi)$ denote active and reactive demand, and $s^{\text{hist}}(\xi)$ denotes optional historical and contextual operating features.

The primal solution is modeled as the network-indexed function

$$x^\star(\xi) = [P_G^\star(\xi), Q_G^\star(\xi), V^\star(\xi), \theta^\star(\xi)]. \tag{22}$$

The dual solution associated with inequality constraints is represented as

$$\mu^\star(\xi) = \left[\mu_G^\star(\xi), \mu_V^\star(\xi), \mu_F^\star(\xi), \mu_u^\star(\xi), \mu_q^\star(\xi)\right], \tag{23}$$

where $\mu_G^\star$, $\mu_V^\star$, $\mu_F^\star$, $\mu_u^\star$, and $\mu_q^\star$ correspond to the generator-limit, voltage-limit, branch-flow, u-constraint, and q-constraint inequality multipliers, respectively.

Through this viewpoint, AC-OPF is not treated as a single fixed-dimensional input–output regression task. Instead, it is formulated as learning a structured operator that maps operating-condition functions over the power network to primal–dual solution functions over buses and branches.

4 Proposed Deep Operator Learning Framework

This section presents the proposed learning framework for approximating the primal–dual AC-OPF solution operator introduced in (20). Consistent with the formulation in the previous section, the goal is to learn the mapping from the operating-condition function $u(\cdot)$ to the optimal primal and dual solution functions. To this end, we combine a DeepONet-based operator architecture with a differentiable dual refinement mechanism and a KKT-informed training objective.

4.1 Primal–Dual DeepONet Architecture

Let $u(\cdot)$ denote the network-indexed operating condition defined in the previous section. In the AC-OPF setting, $u(\cdot)$ may include distributed quantities such as active and reactive power demand, renewable injections, or other exogenous operating inputs associated with buses or other network entities. Depending on the output channel, ξ may correspond to a bus, a generator bus, or a transmission branch. The purpose of the proposed architecture is to approximate the nonlinear operator that maps the input function $u(\cdot)$ to the corresponding optimal primal and dual quantities evaluated at ξ.

To approximate this operator, we adopt a DeepONet architecture. DeepONet is particularly suitable here because AC-OPF is naturally defined over network-indexed functions rather than unstructured vectors. In particular, the operating condition is distributed over physical network entities, and the outputs are likewise attached to buses, generators, or branches. A DeepONet captures this structure through two coordinated components: a branch network that encodes the global input function and a trunk network that encodes the queried evaluation location. Their interaction produces a location-dependent approximation of the desired solution function. This makes DeepONet a better architectural match to the functional structure of AC-OPF than a conventional multilayer perceptron that directly learns a fixed-dimensional vector-to-vector map.

More specifically, the branch network receives sampled observations of the operating-condition function over a set of measurement or sampling locations $\{\xi_i\}_{i=1}^m$. These samples are encoded into a latent representation according to

$$b = B_\theta\big(u(\xi_1), u(\xi_2), \ldots, u(\xi_m)\big), \tag{24}$$

where $B_\theta(\cdot)$ denotes the branch network and θ collects its trainable parameters. The vector b summarizes the global operating condition and serves as a shared latent encoding for all subsequent output evaluations.

For a given query location ξ, the trunk network generates a location-dependent embedding

$$t(\xi) = T_\phi(\xi), \tag{25}$$

where $T_\phi(\cdot)$ denotes the trunk network with trainable parameters ϕ. Whereas the branch network captures the global operating state, the trunk network identifies where in the network the output is being evaluated. This separation between global-context encoding and local-query encoding is a key reason for the suitability of DeepONet in the present setting.

The primal output is then produced through a DeepONet fusion of branch and trunk features. Let p denote the latent width of the corresponding operator head. The predicted primal quantity at query location ξ is written as

$$\hat{x}(\xi) = \sum_{k=1}^{p} b_{x,k}\, t_{x,k}(\xi) + c_x, \tag{26}$$

where $b_{x,k}$ and $t_{x,k}(\xi)$ denote the kth branch and trunk features used for the primal-output channel, and c_x is a learned bias term. In this work, $\hat{x}(\xi)$ represents the predicted primal solution function, whose components may include generator outputs, voltage magnitudes, voltage angles, or branch-flow quantities depending on the queried entity and output head.

In a similar manner, the model predicts raw inequality multipliers associated with the AC-OPF inequality constraints. Since inequality multipliers must satisfy the dual-feasibility condition $\mu \geq 0$ as introduced in (14), the raw multiplier output is passed through a Softplus activation:

$$\tilde{\mu}(\xi) = \text{Softplus}\left(\sum_{k=1}^{p} b_{\mu,k}\, t_{\mu,k}(\xi) + c_\mu\right), \tag{27}$$

where $b_{\mu,k}$ and $t_{\mu,k}(\xi)$ denote latent branch and trunk features for the dual-output channel, and c_μ is a learned bias.

The Softplus activation provides a smooth approximation to the nonnegative projection and therefore promotes, though does not fully enforce, dual feasibility.

Collecting the branch network, trunk network, and output heads, the resulting learned operator is denoted by

$$\Phi_\Theta : u(\cdot) \mapsto (\hat{x}(\xi), \tilde{\mu}(\xi)), \tag{28}$$

where Θ denotes the full set of trainable parameters. The notation in (28) emphasizes that the model maps a distributed operating condition to predicted primal quantities and raw inequality multipliers evaluated at network query locations.

The overall workflow is summarized in Fig. 1. First, the operating condition $u(\cdot)$ is represented as a network-indexed input function and sampled over the power system. These samples are processed by the shared branch network to form a global latent representation. Next, the queried network entity ξ is embedded by the trunk network. The branch and trunk representations are then fused by output-specific DeepONet heads to produce the predicted primal variables and raw inequality multipliers. Finally, the raw dual prediction is passed to a differentiable dual completion layer, which refines the multipliers to improve consistency with the KKT conditions. In this way, the model architecture combines the expressive function-approximation capability of DeepONet with explicit optimization-aware refinement.

4.2 Differentiable Dual Completion Layer

Although the Softplus transformation in (27) promotes nonnegative multiplier predictions, nonnegativity alone is not sufficient for KKT consistency. As discussed in the previous section, the AC-OPF solution must satisfy not only dual feasibility but also complementary slackness and stationarity; see (15)–(19). Direct regression of dual variables often fails to satisfy these structural conditions, especially near active-set transitions where the multipliers may change sharply. To address this issue, we introduce a differentiable dual completion layer that refines the raw multiplier prediction in a constraint-aware manner.

As shown in Fig. 1, the dual completion layer takes as input the predicted primal solution $\hat{x}$ together with the raw multiplier prediction $\tilde{\mu}$ and produces a refined multiplier $\hat{\mu}$. The refinement is guided by the inequality residuals and by a stationarity-aware regularization term so that the final dual estimate is more consistent with the structure of the KKT system.

To define the refinement, recall from the compact AC-OPF formulation that $h(x, u) \leq 0$ collects the inequality constraints. Using the predicted primal solution $\hat{x}$, we define the inequality residual function as

$$s(\xi) = h(\hat{x}, u)(\xi), \tag{29}$$

where $s(\xi) \leq 0$ indicates inequality feasibility at query location ξ. The complementarity condition requires the product between the multiplier and the corresponding inequality residual to vanish, i.e.,

$$\mu(\xi)\, s(\xi) = 0. \tag{30}$$

Thus, a KKT-consistent prediction should lie near one of two regimes: either the constraint is inactive, in which case $s(\xi) < 0$ and $\mu(\xi) = 0$, or the constraint is active, in which case $s(\xi) = 0$ and $\mu(\xi) \geq 0$.

This geometric interpretation is illustrated in Fig. 2, where the admissible primal–dual structure is concentrated near the union of the inactive region and the active boundary.

Formally, given $\hat{x}$ and $\tilde{\mu}$, the refined multiplier is defined as the approximate solution of the following nonnegative optimization problem:

$$\min_{\mu \geq 0} \frac{1}{2} \|\mu - \tilde{\mu}\|_2^2 + \alpha \|\mu \odot h(\hat{x}, u)\|_2^2 + \beta \mathcal{R}_{\text{stat}}(\hat{x}, \mu), \tag{31}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are penalty coefficients, $\odot$ denotes the Hadamard (elementwise) product, and $\mathcal{R}_{\text{stat}}(\hat{x}, \mu)$ is a stationarity-related residual term. The first term keeps the refined multiplier close to the raw network prediction, the second term penalizes complementarity violation, and the third term encourages consistency with the stationarity structure of the AC-OPF Lagrangian introduced in (13).

Rather than solving (31) exactly at each forward pass, we implement the layer using a fixed number of differentiable projected-gradient iterations. Starting from the initialization

$$\mu^{(0)} = \tilde{\mu}, \tag{32}$$

the update at iteration k is given by

$$\mu^{(k+1)} = \Pi_{\mathbb{R}^m_+} \left[\mu^{(k)} - \eta \nabla_\mu \mathcal{J}\left(\mu^{(k)}\right) \right], \tag{33}$$

where $\eta > 0$ is the step size, $\Pi_{\mathbb{R}^m_+}(\cdot)$ denotes projection onto the nonnegative orthant, and m here denotes the dimension of the multiplier vector being refined in the current operator evaluation. The refinement objective $\mathcal{J}(\mu)$ is

$$\mathcal{J}(\mu) = \frac{1}{2} \|\mu - \tilde{\mu}\|_2^2 + \alpha \|\mu \odot h(\hat{x}, u)\|_2^2 + \beta \mathcal{R}_{\text{stat}}(\hat{x}, \mu). \tag{34}$$

After K projected-gradient iterations, the final refined multiplier is defined as

$$\hat{\mu} = \mu^{(K)}. \tag{35}$$

Because all operations in (33) are differentiable almost everywhere, the refinement layer can be embedded directly into end-to-end training. This allows the model to learn raw dual predictions that are not only accurate in a supervised sense but also easier to refine into KKT-consistent multipliers.

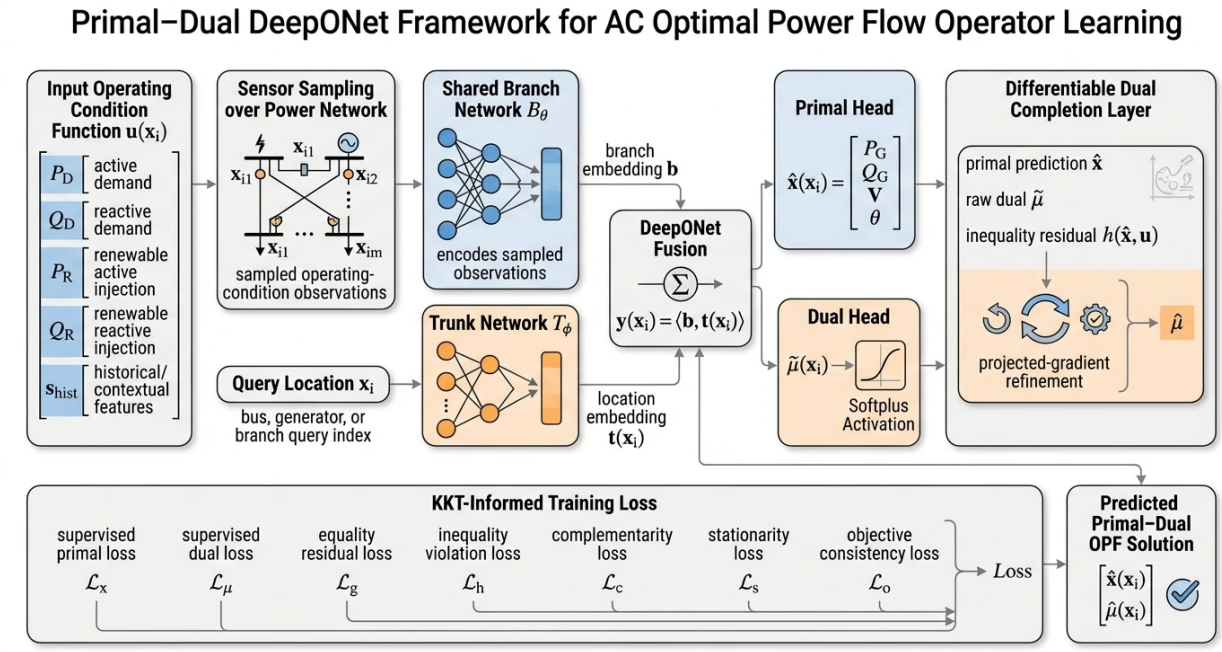


Figure 1: Overview of the proposed primal–dual DeepONet framework for AC-OPF operator learning. The operating-condition function $u(\xi)$ is sampled over the power network and encoded by a shared branch network B_θ. A trunk network T_ϕ embeds the queried network location ξ. The branch and trunk representations are fused through DeepONet operator heads to predict the primal solution $\hat{x}(\xi)$ and the raw inequality multiplier $\tilde{\mu}(\xi)$. The raw dual prediction is then refined.

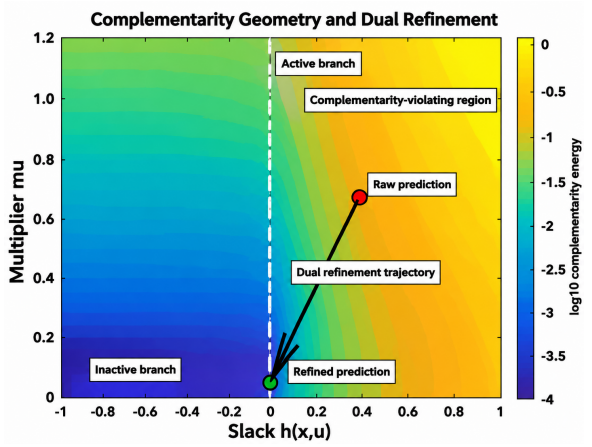


Figure 2: Geometric interpretation of complementarity-aware dual refinement. The refinement step drives the pair (s, μ) toward KKT-consistent regions satisfying inequality feasibility, dual feasibility, and complementary slackness.

4.3 KKT-Informed Operator Loss

The proposed model is trained using a hybrid loss that combines supervised prediction errors with physics-informed and optimization-informed regularizers. This design follows directly from the compact AC-OPF formulation (10)–(12) and the KKT conditions (15)–(19). The overall loss is also summarized in Fig. 1.

Let $\{\xi_j\}_{j=1}^{N_q}$ denote the set of query locations used in a training batch, and let $x^\star(\xi_j)$ and $\mu^\star(\xi_j)$ denote the corresponding reference primal and inequality-dual solutions. The total training objective is defined as

$$\begin{aligned}\mathcal{L}_{\text{total}} = w_x\mathcal{L}_x + w_\mu\mathcal{L}_\mu + w_g\mathcal{L}_g + w_h\mathcal{L}_h \\ + w_c\mathcal{L}_c + w_s\mathcal{L}_s + w_o\mathcal{L}_o,\end{aligned} \tag{36}$$

where $\{w_x, w_\mu, w_g, w_h, w_c, w_s, w_o\}$ are nonnegative scalar weights that balance the relative influence of the individual loss components.

The primal supervision term penalizes the discrepancy between the predicted and reference primal quantities:

$$\mathcal{L}_x = \frac{1}{N_q}\sum_{j=1}^{N_q} \|\hat{x}(\xi_j) - x^\star(\xi_j)\|_2^2 . \tag{37}$$

Similarly, the dual supervision term penalizes the error in the refined multiplier prediction:

$$\mathcal{L}_\mu = \frac{1}{N_q}\sum_{j=1}^{N_q} \|\hat{\mu}(\xi_j) - \mu^\star(\xi_j)\|_2^2 . \tag{38}$$

To promote feasibility with respect to the AC-OPF constraints, we include an equality-residual loss and an inequality-violation loss. The equality-residual loss is defined by

$$\mathcal{L}_g = \|g(\hat{x}, u)\|_2^2 , \tag{39}$$

where $g(\hat{x}, u)$ collects the power-balance and other equality residuals evaluated at the predicted primal solution. The inequality-violation loss penalizes only positive violations of the inequality constraints:

$$\mathcal{L}_h = \left\|\max\left(0, h(\hat{x}, u)\right)\right\|_2^2 . \tag{40}$$

Thus, $\mathcal{L}_g$ encourages satisfaction of the equality constraints, whereas $\mathcal{L}_h$ discourages violations of generator, voltage, and line-flow limits.

Table 2: Training configuration and model hyperparameters used in the experiments.

Item	Configuration
Optimizer	Adam
Initial learning rate	1×10^{-3}
Learning-rate schedule	Reduce-on-plateau, decay 0.5, patience 10 epochs
Batch size	64
Maximum epochs	250
Early stopping patience	25 epochs
Activation function	ReLU
Branch network	4 hidden layers, width 256
Trunk network	4 hidden layers, width 256
Dual head	2 hidden layers, width 128
Normalization	Feature-wise standardization
Dual completion steps	3 projected-gradient steps
Dual completion step size	0.05
Hardware	Single GPU (identical for all learned baselines)

To encourage KKT consistency beyond basic feasibility, we further include complementarity and stationarity penalties. The complementarity loss is

$$\mathcal{L}_c = \|\hat{\mu} \odot h(\hat{x}, u)\|_2^2 , \tag{41}$$

which penalizes violations of the complementary-slackness condition (19). The stationarity loss is defined as

$$\mathcal{L}_s = \left\| \nabla_x \mathcal{L}\big(\hat{x}, \hat{\lambda}, \hat{\mu}; u\big) \right\|_2^2 , \tag{42}$$

where $\mathcal{L}(\cdot)$ denotes the AC-OPF Lagrangian defined in (13), and $\hat{\lambda}$ denotes the equality-constraint multiplier used during training. If equality multipliers are explicitly predicted by the model, then $\hat{\lambda}$ is taken directly from the network output. Otherwise, $\hat{\lambda}$ may be estimated approximately from the stationarity equations, for example through a least-squares solve. This point should be implemented consistently with the training setup adopted in the experiments.

Finally, to encourage economic consistency with the true OPF objective, we include an objective-matching term:

$$\mathcal{L}_o = |f(\hat{x}; u) - f(x^\star; u)|^2 , \tag{43}$$

where $f(x; u)$ is the compact objective introduced in (10). This term promotes agreement not only in the predicted variables themselves but also in the resulting operating cost.

Taken together, the losses in (36) encourage the learned operator to produce outputs that are simultaneously accurate, feasible, and consistent with the first-order optimality structure of AC-OPF. Supervised terms anchor the model to reference OPF solutions, feasibility terms enforce physical validity, and KKT-based penalties guide the network toward optimization-consistent primal–dual predictions.

5 Training and Inference

5.1 Training Procedure

The proposed framework is trained using a dataset of AC-OPF instances generated offline. Each sample contains distributed operating conditions and the corresponding optimal primal solution. When available, the associated optimal inequality dual variables are also included as training targets. The objective is to learn the mapping from network operating conditions to the corresponding primal–dual AC-OPF solution.

To address the nonlinear nature of AC-OPF and the additional sensitivity introduced by the KKT-informed objectives, the model components are trained progressively rather than initialized and optimized simultaneously. This staged strategy improves convergence stability and allows the primal predictor, dual predictor, and optimization-informed components to be incorporated sequentially. Crucially, each stage is treated as an independent optimization subproblem: the training criteria—including maximum epochs and early stopping patience—are applied individually to each stage to trigger the transition to the next. This prevents unnecessary computation once a specific component has reached a stable local minimum. The complete training configuration, including the optimizer, learning-rate schedule, stopping criteria, network architecture, normalization procedure, and dual-completion settings, is reported in Table 2.

5.1.1 Stage 1: Primal Pretraining

The first stage focuses only on learning the primal solution manifold. In this phase, the branch and trunk networks are trained using the previously defined primal supervision term so that the model first captures the dominant mapping from operating conditions to optimal voltage and generation variables. This initialization provides a stable starting point for the later physics-aware stages.

5.1.2 Stage 2: Feasibility-Aware Training

After obtaining a reasonable primal initialization, the equality and inequality residual terms are activated. These losses encourage the predicted solution to satisfy the AC power-flow equations and operational constraints more closely. As a result, the network moves from purely data-driven regression toward physically meaningful AC-OPF predictions.

5.1.3 Stage 3: Dual Learning

Next, the dual prediction head is introduced and trained using the previously defined dual supervision term. Learning dual variables is more challenging than learning primal variables because the dual solution depends strongly on the active constraint set and may vary sharply near binding transitions. Nevertheless, this stage is important for capturing marginal sensitivities and congestion behavior.

5.1.4 Stage 4: KKT-Informed Refinement

Once the model provides stable and meaningful primal and dual estimates, the complementarity, stationarity, and objective-consistency losses are activated. At this stage, the differentiable dual completion layer is incorporated directly into the computational graph so that the network is refined not only toward label accuracy, but also toward first-order KKT optimality consistency.

Table 3: Progressive staged training procedure used for the proposed framework.

Stage	Name	Activated components	Main objective
1	Primal pretraining	Branch + trunk networks, primal supervision	Learn the dominant mapping from operating conditions to primal OPF variables
2	Feasibility-aware training	Stage 1 + equality/inequality residual losses	Improve physical feasibility with respect to AC power-flow and operating limits
3	Dual learning	Stage 2 + dual prediction head + dual supervision	Learn inequality-multiplier behavior and congestion-related sensitivities
4	KKT-informed refinement	Stage 3 + dual completion layer	Align predictions with first-order optimality structure
5	End-to-end fine-tuning	All components jointly optimized	Balance accuracy, feasibility, and optimization consistency

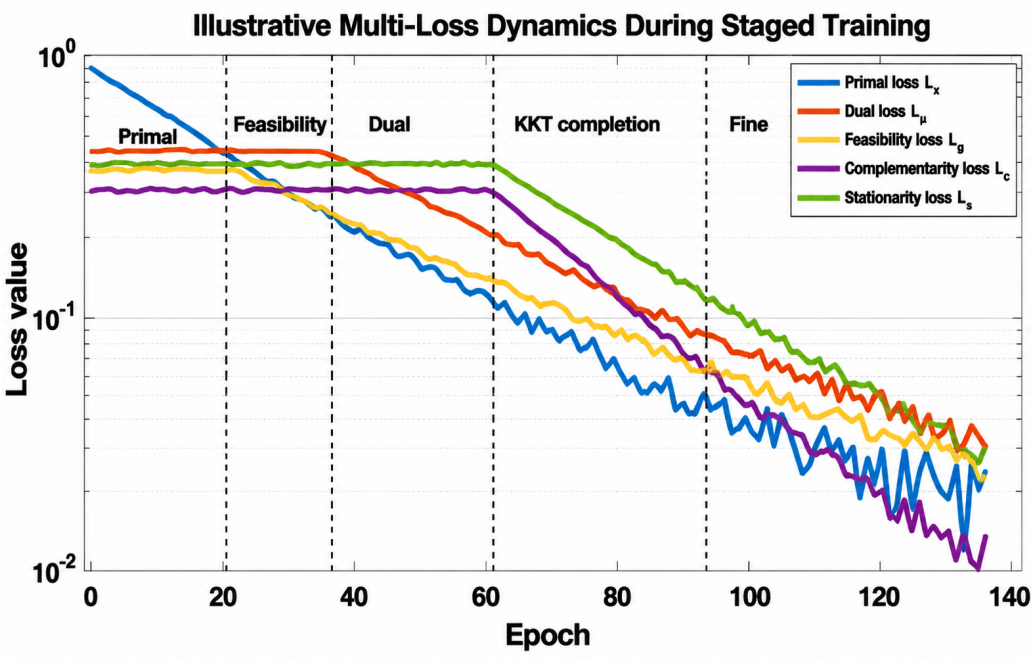


Figure 3: Illustrative staged training dynamics of the proposed framework. The model is first pretrained for primal prediction, followed by feasibility-aware learning, dual supervision, KKT-informed refinement, and final end-to-end fine-tuning. The staged activation of losses improves convergence stability and leads to lower residuals in both prediction and first-order optimality conditions.

In implementation, the dual completion module performs a small, fixed number of projected-gradient steps, configured as $K = 3$ updates with a step size of $\eta = 0.05$. This choice is guided by the convergence dynamics of the first-order optimality conditions: the projected updates rapidly contract the complementarity residual r_c and inequality constraint violations within the initial few steps, after which additional iterations yield diminishing accuracy returns while linearly increasing the computational burden. Moreover, the moderate step size $\eta = 0.05$ ensures smooth, monotonic residual reduction without introducing oscillations around active constraint boundaries. Consequently, this fixed update budget strikes an effective balance between strict structural alignment (coupling predicted multipliers with active physical constraints) and computational efficiency during end-to-end training and inference.

5.1.5 Stage 5: End-to-End Fine-Tuning

In the final stage, all components are jointly fine-tuned in an end-to-end manner using the full set of previously introduced training objectives. As with the preceding stages, the optimization for Stage 5 is governed by its own dedicated convergence criteria (as specified in Table 2), ensuring that the final joint refinement is thoroughly optimized for stability and accuracy. This final refinement balances prediction accuracy, physical feasibility, and optimization consistency, leading to improved generalization on unseen operating conditions.

Table 3 and Fig. 3 summarize the staged training strategy. The sequential activation of supervision, feasibility, and KKT-based objectives improves convergence stability and reduces the risk of poor local minima during optimization.

Table 4: Dataset summary and train/validation/test split for each benchmark system.

System	Buses	Generators	Branches	Total scenarios	Train / Val / Test	Load perturbation range
IEEE 30	30	6	41	12,000	8,400 / 1,800 / 1,800	$\pm 20\%$
IEEE 118	118	54	186	18,000	12,600 / 2,700 / 2,700	$\pm 20\%$
IEEE 300	300	69	411	24,000	16,800 / 3,600 / 3,600	$\pm 20\%$

5.2 Inference Procedure

After training, the learned operator is used as a fast surrogate for AC-OPF. Given a new operating-condition function $u(\cdot)$, the model predicts the associated primal and dual solution functions over the network by evaluating the trained operator at the desired query location ξ. In practical terms, these query locations correspond to buses, generators, or branches depending on the target variable.

To improve robustness, the raw dual outputs are passed through the differentiable dual completion layer, which performs a small fixed number of projected updates to enforce nonnegativity and improve complementarity consistency. This refinement is lightweight and does not require solving the full nonlinear optimization problem.

Consequently, inference consists only of a forward pass through the trained operator together with a small refinement step, avoiding the repeated Newton iterations and matrix factorizations required by conventional AC-OPF solvers. This makes the proposed framework well suited for applications requiring rapid repeated evaluations, such as real-time dispatch support, contingency screening, and large-scale scenario analysis.

6 Experiments

6.1 Experimental Setup

The proposed framework was evaluated on three standard IEEE transmission-system benchmarks: the IEEE 30-bus, 118-bus, and 300-bus systems. These test cases provide a representative range of medium- and large-scale AC-OPF settings and enable evaluation under increasing network complexity. For each system, operating scenarios were generated by perturbing active and reactive demand around the nominal load profile, while the network topology, line parameters, generator limits, voltage bounds, and other operational constraints were kept fixed. This setting captures realistic load variability while preserving the physical structure of each benchmark system.

For each scenario index s and load bus i, the active and reactive demands were sampled according to

$$P_{D,i}^{(s)} = P_{D,i}^{0}\left(1+\delta_{P,i}^{(s)}\right), \qquad Q_{D,i}^{(s)} = Q_{D,i}^{0}\left(1+\delta_{Q,i}^{(s)}\right), \tag{44}$$

where $P_{D,i}^{0}$ and $Q_{D,i}^{0}$ denote the nominal active and reactive demand values, respectively. The perturbation factors $\delta_{P,i}^{(s)}$ and $\delta_{Q,i}^{(s)}$ were independently sampled from a uniform distribution over $[-0.2, 0.2]$. The resulting scenarios include both lightly and heavily loaded operating conditions, providing a broad test range for assessing prediction accuracy, feasibility, and optimization consistency.

Each sampled operating scenario was solved using a conventional AC-OPF solver to generate the reference primal and dual labels. The resulting datasets were divided into training, validation, and test subsets using a $70/15/15$ split. To prevent data leakage, all normalization statistics were computed using only the training subset and then applied unchanged to the validation and test subsets. The dataset sizes and corresponding splits for the IEEE 30-, 118-, and 300-bus systems are summarized in Table 4.

All experiments were performed on the same computing platform to ensure consistent runtime comparisons across the proposed framework and all baseline methods. The hardware consisted of an Intel Core i7-3370 CPU, 16 GB RAM, and an AMD Radeon RX 590 GPU with 8 GB memory.

6.2 Baselines

To assess the benefit of the proposed primal–dual operator-learning approach, it was compared against representative learning-based surrogates:

- **MLP Primal:** a conventional multilayer perceptron trained to directly regress the primal AC-OPF variables;
- **DeepONet Primal:** an operator-learning baseline that predicts only primal variables without dual refinement;
- **Proposed Method:** the full primal–dual DeepONet with differentiable dual completion and KKT-informed training.

This comparison isolates the contribution of operator learning, dual prediction, and explicit optimality regularization.

6.3 Evaluation Metrics

Performance was evaluated from four complementary perspectives: prediction accuracy, feasibility, optimization consistency, and economic quality.

6.3.1 Prediction Accuracy

Primal and dual errors were quantified using root-mean-square error (RMSE):

$$\mathrm{RMSE}_x = \sqrt{\frac{1}{N_q}\sum_{j=1}^{N_q}\|\hat{x}(\xi_j) - x^{\star}(\xi_j)\|_2^2}, \tag{45}$$

$$\mathrm{RMSE}_\mu = \sqrt{\frac{1}{N_q}\sum_{j=1}^{N_q}\|\hat{\mu}(\xi_j) - \mu^{\star}(\xi_j)\|_2^2}, \tag{46}$$

Table 5: Definition of evaluation metrics and parameter units reported in the experiments.

Metric	Symbol	Unit	Interpretation
Primal root-mean-square error	RMSE_x	p.u.	Average prediction error of primal variables (voltages and power)
Dual root-mean-square error	RMSE_μ	p.u.	Average prediction error of inequality multipliers μ
Equality residual	r_g	p.u.	Violation of AC power-flow equality constraints $g(x,u)=0$
Inequality residual	r_h	p.u.	Violation of operational inequality constraints $h(x,u)\le 0$
Complementarity residual	r_c	p.u.	Consistency between multipliers μ and inequality slacks $h(x,u)$
Stationarity residual	r_s	p.u.	Norm of the Lagrangian gradient $\nabla_x\mathcal{L}$
Feasibility rate	–	%	Percentage of test scenarios satisfying prescribed tolerances
Objective gap	$\overline{\mathrm{Gap}}_{\mathrm{obj}}$	%	Average percentage cost gap relative to the AC-OPF optimum
Inference runtime	Time	ms	Average forward-pass evaluation time per scenario

where N_q denotes the number of evaluated query locations in the network.

6.3.2 Feasibility Metrics

Constraint satisfaction was quantified using the equality residual

$$r_g = \|g(\hat{x},u)\|_2, \tag{47}$$

and the inequality violation metric

$$r_h = \|\max(0,h(\hat{x},u))\|_2. \tag{48}$$

Smaller values of r_g and r_h indicate closer adherence to the AC power-flow equations and operational limits.

In addition, a scenario-level feasibility rate was computed to quantify the percentage of test cases for which the predicted solution satisfies the AC-OPF constraints within a prescribed tolerance. Let ε_g and ε_h denote the tolerances for equality and inequality feasibility, respectively; in this study, we set $\varepsilon_g = 10^{-5}$ p.u. and $\varepsilon_h = 10^{-5}$ p.u. The feasibility indicator for scenario s is defined as

$$\mathbb{I}_{\mathrm{feas}}^{(s)} = \begin{cases} 1, & \|g(\hat{x}^{(s)},u^{(s)})\|_2 \le \varepsilon_g \\ & \text{and } \|\max(0,h(\hat{x}^{(s)},u^{(s)}))\|_\infty \le \varepsilon_h \\ 0, & \text{otherwise,} \end{cases} \tag{49}$$

and the feasibility percentage is then given by

$$\text{Feasibility Rate } (\%) = 100 \times \frac{1}{N_{\mathrm{test}}} \sum_{s=1}^{N_{\mathrm{test}}} \mathbb{I}_{\mathrm{feas}}^{(s)}. \tag{50}$$

6.3.3 Optimality Metrics

To assess whether the predictions satisfy first-order optimality structure, two KKT-based residuals were computed:

$$r_c = \|\hat{\mu} \odot h(\hat{x},u)\|_2, \tag{51}$$

$$r_s = \|\nabla_x \mathcal{L}(\hat{x},\hat{\lambda},\hat{\mu};u)\|_2. \tag{52}$$

Here, r_c measures complementarity consistency between dual variables and inequality constraints, while r_s measures stationarity of the predicted solution with respect to the Lagrangian.

6.3.4 Objective Gap

To evaluate the economic quality of the predicted dispatch, the percentage objective gap relative to the reference AC-OPF optimum was also measured. For each test scenario s, the gap is defined as

$$\mathrm{Gap}_{\mathrm{obj}}^{(s)}\ (\%) = 100 \times \frac{f(\hat{x}^{(s)},u^{(s)}) - f(x^{\star(s)},u^{(s)})}{f(x^{\star(s)},u^{(s)})}. \tag{53}$$

The average percentage objective gap over the test set is then reported as

$$\overline{\mathrm{Gap}}_{\mathrm{obj}}\ (\%) = \frac{1}{N_{\mathrm{test}}} \sum_{s=1}^{N_{\mathrm{test}}} \mathrm{Gap}_{\mathrm{obj}}^{(s)}. \tag{54}$$

A smaller objective gap indicates that the predicted operating point remains closer in cost to the true optimum. Table 5 defines our key evaluation metrics.

7 Results and Discussion

7.1 Overall Performance Comparison

Tables 6 and 7 compare the proposed primal–dual DeepONet with the baseline models on the IEEE 30-bus, 118-bus, and 300-bus systems. The results show that the proposed framework consistently achieves the strongest overall performance in primal accuracy, dual accuracy, feasibility, KKT consistency, objective quality, and inference efficiency.

As the system size increases, the advantage of the proposed method becomes more evident. This trend indicates that representing AC-OPF as a network-indexed operator-learning problem is more effective than treating it as a fixed-dimensional regression task. The improvement is especially clear in the feasibility and KKT-related metrics, suggesting that the combination of operator learning, dual completion, and KKT-informed training better captures the nonlinear coupling among voltage states, generator outputs, branch constraints, and multipliers.

The proposed method requires slightly higher inference time than the MLP and primal-only DeepONet baselines. However, the runtime remains below 50 ms for all tested systems, including the IEEE 300-bus case. Therefore, the additional computational cost is modest relative to the gains in feasibility, optimality, and primal–dual consistency.

Table 6: Performance comparison on IEEE benchmark systems in terms of primal and dual prediction accuracy, KKT-related residuals, and inference time. Lower values are better for all metrics except feasibility rate in Table 7.

System	Method	RMSE_x	RMSE_μ	r_g	r_h	r_c	r_s	Time
IEEE 30	MLP Primal	6.35e-4	3.42e-3	3.86e-3	2.94e-3	2.71e-3	5.84e-3	18.6 ms
	DeepONet Primal	4.58e-4	3.08e-3	3.41e-3	2.66e-3	2.38e-3	5.12e-3	24.8 ms
	Proposed	**4.51e-4**	**3.84e-4**	**6.08e-4**	**5.31e-4**	**2.14e-4**	**4.96e-4**	37.2 ms
IEEE 118	MLP Primal	8.02e-4	4.86e-3	5.44e-3	4.17e-3	3.62e-3	7.95e-3	19.7 ms
	DeepONet Primal	5.86e-4	4.41e-3	4.92e-3	3.81e-3	3.24e-3	7.08e-3	26.9 ms
	Proposed	**5.79e-4**	**4.92e-4**	**7.65e-4**	**6.58e-4**	**2.86e-4**	**6.21e-4**	40.3 ms
IEEE 300	MLP Primal	1.11e-3	6.92e-3	7.83e-3	5.94e-3	5.08e-3	1.13e-2	21.4 ms
	DeepONet Primal	8.02e-4	6.35e-3	7.12e-3	5.46e-3	4.61e-3	1.02e-2	29.8 ms
	Proposed	**7.94e-4**	**6.88e-4**	**1.01e-3**	**8.95e-4**	**3.92e-4**	**7.74e-4**	44.7 ms

Table 7: Feasibility rate and average objective gap on IEEE benchmark systems. Higher feasibility rates and lower objective gaps indicate better practical solution quality.

System	Method	Feasibility Rate (%)	Objective Gap (%)
IEEE 30	MLP Primal	91.4	0.44
	DeepONet Primal	95.2	0.26
	Proposed	**99.7**	**0.09**
IEEE 118	MLP Primal	91.0	0.67
	DeepONet Primal	94.8	0.58
	Proposed	**99.5**	**0.14**
IEEE 300	MLP Primal	90.7	0.81
	DeepONet Primal	94.3	0.51
	Proposed	**99.3**	**0.20**

7.2 Primal Accuracy and Feasibility

Table 6 shows that the proposed method achieves the lowest primal prediction error on all benchmark systems. Compared with MLP Primal, the proposed framework reduces RMSE_x by approximately 29.0%, 27.8%, and 28.5% on the IEEE 30-bus, 118-bus, and 300-bus systems, respectively. The improvement over the primal-only DeepONet is smaller but consistent, indicating that KKT-informed learning improves accuracy beyond the benefit obtained from operator learning alone.

The feasibility-related residuals show a stronger improvement. On the IEEE 300-bus system, the equality residual r_g decreases from 7.83×10^{-3} for MLP Primal and 7.12×10^{-3} for DeepONet Primal to 1.01×10^{-3}. Similarly, the inequality residual r_h is reduced to 8.95×10^{-4}, substantially lower than both baselines. These reductions indicate that the predicted operating points remain closer to the AC-feasible region.

The stationarity residual also decreases significantly. For IEEE 300, r_s is reduced from 1.13×10^{-2} for MLP Primal and 1.02×10^{-2} for DeepONet Primal to 7.74×10^{-4}. This result suggests that the predictions are not only close to the reference labels, but also more consistent with the first-order optimality structure of AC-OPF.

7.3 Dual Prediction and Complementarity

The proposed method provides a substantial improvement in dual-variable prediction. As shown in Table 6, RMSE_μ is reduced by approximately 88.8%, 89.9%, and 90.1% relative to MLP Primal on the IEEE bus, 118-bus, and 300-bus systems, respectively. This improvement is important because dual variables are sensitive to active-set changes and are generally harder to learn than primal variables.

Complementarity performance follows the same pattern. The proposed framework reduces r_c from 2.71×10^{-3} to 2.14×10^{-4} on IEEE 30, from 3.62×10^{-3} to 2.86×10^{-4} on IEEE 118, and from 5.08×10^{-3} to 3.92×10^{-4} on IEEE 300. These results show that the predicted multipliers are more compatible with the active constraints.

This improvement is mainly attributed to the differentiable dual-completion layer and the KKT-informed loss. Unlike direct dual regression, the proposed approach explicitly encourages nonnegativity and complementarity, reducing inconsistencies between predicted multipliers and constraint activity.

7.4 Feasibility Rate and Objective Quality

Table 7 reports the scenario-level feasibility rate and average objective gap. The proposed framework achieves the highest feasibility rate on all systems: 99.7% on IEEE 30, 99.5% on IEEE 118, and 99.3% on IEEE 300. The improvement over MLP Primal reaches 8.6 percentage points on IEEE 300, while the gain over DeepONet Primal reaches 5.0 percentage points.

The objective-gap results also favor the proposed method. The average gap is reduced to 0.09%, 0.14%, and

Table 8: Performance and computational cost under varying dual refinement iterations (K) and step sizes (η) on the IEEE 118-bus test set.

K	Step Size (η)	RMSE_x	Residual r_h	Residual r_c	Feasibility (%)	Latency (ms)
0	—	5.86×10^{-4}	2.14×10^{-3}	1.87×10^{-3}	95.1	27.2
1	0.05	5.82×10^{-4}	1.25×10^{-3}	8.94×10^{-4}	97.4	31.6
2	0.05	5.80×10^{-4}	8.12×10^{-4}	4.52×10^{-4}	98.9	35.9
3	0.01	5.81×10^{-4}	9.45×10^{-4}	5.12×10^{-4}	98.2	40.2
3 (Proposed)	**0.05**	$\mathbf{5.79 \times 10^{-4}}$	$\mathbf{6.58 \times 10^{-4}}$	$\mathbf{2.86 \times 10^{-4}}$	**99.5**	**40.3**
3	0.10	5.80×10^{-4}	7.32×10^{-4}	3.49×10^{-4}	98.8	40.4
5	0.05	5.79×10^{-4}	6.41×10^{-4}	2.74×10^{-4}	99.6	49.1

Table 9: Robustness and Generalization Benchmark Across Correlated, Temporal, and Extreme Out-of-Distribution Scenarios (IEEE 118-Bus System)

Testing Regime	Model Architecture	RMSE_x	RMSE_μ	$\mathrm{r_h}$	$\mathrm{r_c}$	Feasibility (%)	Obj. Gap (%)
Regime 1: Baseline *(Uniform ±20%)*	MLP Primal	8.02×10^{-4}	4.86×10^{-3}	4.17×10^{-3}	3.62×10^{-3}	91.0	0.67
	DeepONet Primal	5.86×10^{-4}	4.41×10^{-3}	3.81×10^{-3}	3.24×10^{-3}	94.8	0.58
	Proposed	$\mathbf{5.79 \times 10^{-4}}$	$\mathbf{4.92 \times 10^{-4}}$	$\mathbf{6.58 \times 10^{-4}}$	$\mathbf{2.86 \times 10^{-4}}$	**99.5**	**0.14**
Regime 2: Correlated *(Gaussian Copula Covariance)*	MLP Primal	1.14×10^{-3}	6.23×10^{-3}	5.89×10^{-3}	5.12×10^{-3}	86.4	0.94
	DeepONet Primal	7.94×10^{-4}	5.58×10^{-3}	4.95×10^{-3}	4.38×10^{-3}	91.2	0.76
	Proposed	$\mathbf{6.12 \times 10^{-4}}$	$\mathbf{5.64 \times 10^{-4}}$	$\mathbf{7.42 \times 10^{-4}}$	$\mathbf{3.41 \times 10^{-4}}$	**98.8**	**0.19**
Regime 3: Diurnal Temporal *(24-h Trajectory + Ramps)*	MLP Primal	1.32×10^{-3}	7.05×10^{-3}	6.74×10^{-3}	5.89×10^{-3}	83.1	1.15
	DeepONet Primal	8.65×10^{-4}	6.12×10^{-3}	5.62×10^{-3}	4.95×10^{-3}	88.9	0.89
	Proposed	$\mathbf{6.48 \times 10^{-4}}$	$\mathbf{6.15 \times 10^{-4}}$	$\mathbf{8.19 \times 10^{-4}}$	$\mathbf{3.88 \times 10^{-4}}$	**98.1**	**0.24**
Regime 4: Extreme OOD *(±35% to ±40% Load Surges)*	MLP Primal	2.48×10^{-3}	1.29×10^{-2}	1.42×10^{-2}	1.18×10^{-2}	71.3	2.62
	DeepONet Primal	1.56×10^{-3}	9.84×10^{-3}	1.06×10^{-2}	8.74×10^{-3}	79.5	1.84
	Proposed	$\mathbf{8.73 \times 10^{-4}}$	$\mathbf{8.92 \times 10^{-4}}$	$\mathbf{1.24 \times 10^{-3}}$	$\mathbf{5.93 \times 10^{-4}}$	**95.4**	**0.41**

Note: All metrics are evaluated without retraining any model to measure zero-shot distributional robustness.

0.20% on the IEEE 30-bus, 118-bus, and 300-bus systems, respectively. These values are lower than both baselines across all test cases, showing that the proposed predictions are closer to the economic optimum.

These results are practically relevant because low prediction error alone does not guarantee a usable OPF solution. A high-quality surrogate should produce outputs that are accurate, feasible, and economically near-optimal. The proposed framework improves all three aspects simultaneously.

7.5 Runtime and Scalability

The proposed method has higher inference time than the two baselines because of the additional primal–dual prediction and dual-completion steps. However, the increase remains limited. On IEEE 300, inference time rises from 21.4 ms for MLP Primal and 29.8 ms for DeepONet Primal to 44.7 ms for the proposed method.

This runtime remains suitable for repeated-query applications such as contingency screening, scenario evaluation, and real-time decision support. Compared with conventional AC-OPF solvers, which require iterative nonlinear optimization, the proposed framework provides fast inference while maintaining stronger feasibility and optimality properties than the baseline learning models.

The results also indicate favorable scaling behavior. The largest gains appear on the IEEE 118- and 300-bus systems, where the spatial coupling among buses, generators, and branches is more complex. This supports the use of operator learning and primal–dual KKT alignment for larger AC-OPF problems.

7.6 Sensitivity Analysis of Dual Refinement Parameters (K and η)

To systematically evaluate the trade-off between refinement depth, step size, constraint consistency, and computational overhead, an extensive ablation study was conducted on the IEEE 118-bus benchmark system. We evaluated varying dual refinement steps $K \in \{0, 1, 2, 3, 5\}$ and step sizes $\eta \in \{0.01, 0.05, 0.10\}$. Here, $K = 0$ denotes direct inference without the projected-gradient layer, while $K = 3$ with $\eta = 0.05$ represents the proposed baseline configuration. Table 8 summarizes the primal prediction accuracy (RMSE_x), constraint violation residuals (r_h, r_c), AC feasibility rate, and average per-instance inference latency.

As reported in Table 8, increasing the projected-gradient iterations from $K = 0$ to $K = 3$ (at $\eta = 0.05$) yields substantial improvements in physical and optimality consistency. Specifically, the complementarity residual r_c decreases by nearly an order of magnitude (from 1.87×10^{-3} to 2.86×10^{-4}), the inequality residual r_h contracts by over 69% (from 2.14×10^{-3} to 6.58×10^{-4}), and the AC feasibility rate improves from 95.1% to 99.5%.

Regarding the step size η, when $K = 3$ is fixed:

- A conservative step size ($\eta = 0.01$) proves insufficient to contract boundary violations within three steps, achieving 98.2% feasibility with higher residual levels ($r_c = 5.12 \times 10^{-4}$).
- An aggressive step size ($\eta = 0.10$) induces minor os-

Table 10: Performance and Generalization Under Line Contingencies ($N-1$, $N-2$) and Topology Variations (IEEE 118-Bus System)

Topological Regime	Model Architecture	$\mathbf{RMSE}_x$	$\mathbf{RMSE}_\mu$	$\mathbf{r_h}$	$\mathbf{r_c}$	Feasibility (%)	Obj. Gap (%)
Nominal Base Grid *(Fixed Topology)*	MLP Primal	8.02×10^{-4}	4.86×10^{-3}	4.17×10^{-3}	3.62×10^{-3}	91.0	0.67
	DeepONet Primal	5.86×10^{-4}	4.41×10^{-3}	3.81×10^{-3}	3.24×10^{-3}	94.8	0.58
	Proposed Framework	$\mathbf{5.79 \times 10^{-4}}$	$\mathbf{4.92 \times 10^{-4}}$	$\mathbf{6.58 \times 10^{-4}}$	$\mathbf{2.86 \times 10^{-4}}$	**99.5**	**0.14**
Critical $N-1$ Contingency *(Zero-Shot Base Operator)*	MLP Primal	2.84×10^{-3}	1.41×10^{-2}	1.52×10^{-2}	1.34×10^{-2}	64.2	2.95
	DeepONet Primal	1.89×10^{-3}	1.08×10^{-2}	1.15×10^{-2}	9.82×10^{-3}	74.8	2.10
	Proposed (Zero-Shot)	$\mathbf{1.04 \times 10^{-3}}$	$\mathbf{1.12 \times 10^{-3}}$	$\mathbf{1.58 \times 10^{-3}}$	$\mathbf{6.84 \times 10^{-4}}$	**84.6**	**0.68**
Critical $N-1$ Contingency *(Topology-Aware / Adapted)*	MLP Primal (Fine-tuned)	1.25×10^{-3}	6.75×10^{-3}	6.14×10^{-3}	5.48×10^{-3}	85.3	1.02
	DeepONet Primal (Adapted)	8.12×10^{-4}	5.84×10^{-3}	5.12×10^{-3}	4.62×10^{-3}	90.5	0.81
	Proposed (Topology-Aware)	$\mathbf{6.24 \times 10^{-4}}$	$\mathbf{5.81 \times 10^{-4}}$	$\mathbf{7.65 \times 10^{-4}}$	$\mathbf{3.52 \times 10^{-4}}$	**98.2**	**0.28**
Extreme $N-2$ / Reconfiguration *(Multi-Branch Outages)*	MLP Primal	3.62×10^{-3}	1.88×10^{-2}	1.94×10^{-2}	1.72×10^{-2}	56.4	4.18
	DeepONet Primal	2.41×10^{-3}	1.42×10^{-2}	1.48×10^{-2}	1.29×10^{-2}	68.2	3.05
	Proposed (Topology-Aware)	$\mathbf{8.96 \times 10^{-4}}$	$\mathbf{9.15 \times 10^{-4}}$	$\mathbf{1.28 \times 10^{-3}}$	$\mathbf{6.14 \times 10^{-4}}$	**95.1**	**0.46**

Note: $N-1$ and $N-2$ contingency evaluations target critical corridor line outages without relaxing AC power flow feasibility tolerances.

Table 11: Operational Feasibility Restoration and Latency Comparison Across Solver Warm-Starting Schemes (IEEE 118-Bus System)

Resolution Strategy	Initial Feas. (%)	Certified Feas. (%)	Avg. Solver Iterations	Solver Time (ms)	Total Time (ms)
Cold-Start IPM (IPOPT Baseline)	0.0	100.0	23.16 ± 4.82	248.50 ± 38.10	248.50 ± 38.10
Naive L_2-Norm Primal Projection	91.0	96.4	—	4.15 ± 0.62	7.45 ± 0.85
Power-Flow Recovery (Newton-Raphson)	94.8	98.1	3.62 ± 1.10	9.80 ± 1.40	13.20 ± 1.65
Proposed (Raw Prediction)	**99.5**	**99.5**	**—**	**—**	$\mathbf{3.28 \pm 0.42}$
Proposed + Primal Warm-Start IPM	99.5	100.0	5.12 ± 1.45	38.40 ± 8.20	41.68 ± 8.35
Proposed + Primal–Dual Warm-Start IPM	**99.5**	**100.0**	$\mathbf{2.84 \pm 0.76}$	$\mathbf{15.14 \pm 3.12}$	$\mathbf{18.42 \pm 3.25}$

Note: Total time accounts for model forward inference (3.28 ms) plus the iterative restoration solver time on identical hardware.

cillatory behavior near active inequality constraints, degrading feasibility slightly to 98.8%.

- The moderate value $\eta = 0.05$ achieves the lowest residuals and the highest feasibility rate (99.5%), ensuring smooth, monotonic convergence.

Furthermore, extending the iterations beyond $K = 3$ exhibits a distinct plateau with diminishing returns: increasing K to 5 yields only a marginal feasibility gain ($+0.1\%$) and negligible residual drop (r_h to 6.41×10^{-4}, r_c to 2.74×10^{-4}), while increasing execution latency by approximately 22% (49.1 ms vs. 40.3 ms). Therefore, configuring $K = 3$ with $\eta = 0.05$ represents the optimal Pareto trade-off between strict KKT compliance and real-time execution capability on large-scale transmission networks.

7.7 Robustness Under Correlated, Temporal, and Extreme Out-of-Distribution Scenarios

To assess real-world operational viability, we evaluate the models trained on baseline uniform perturbations under three challenging zero-shot generalization regimes on the IEEE 118-bus network, without model retraining or hyperparameter re-tuning. First, in the *Spatially Correlated Regime*, active and reactive load variations are generated using a Gaussian copula with spatial exponential covariance matrix $\Sigma_{ij} = \exp(-d_{ij}/\ell)$, where d_{ij} represents the electrical shortest-path distance between buses i and j, and ℓ denotes the spatial correlation length scale over $[\pm 20\%]$ perturbation bounds. Second, in the *Diurnal Temporal Regime*, we simulate a 24-hour sequential load profile featuring morning and evening peak demands coupled with intermittent solar and wind ramping variations ($\pm 25\%$). Third, in the *Extreme Out-of-Distribution (OOD) Stress Regime*, severe system-wide stress scenarios are introduced where load perturbations reach $[\pm 35\%, \pm 40\%]$ of nominal ratings, pushing branch power flows and voltage bounds directly against their critical thermal and operational limits.

As reported in Table 9, conventional supervised baselines experience severe degradation when transitioning from uniform to correlated, temporal, and extreme OOD regimes. For instance, in the extreme OOD regime, the feasibility rate of the standard MLP drops sharply from 91.0% to 71.3%, while its objective suboptimality increases to 2.62%. Similarly, the DeepONet Primal model suffers a feasibility drop to 79.5%. This breakdown occurs because purely data-driven models lack structural awareness of the operational manifold under unseen distributional tails.

In contrast, the proposed Primal–Dual DeepONet demonstrates superior resilience across all test regimes. In the spatially correlated and diurnal regimes, the proposed model sustains feasibility rates of 98.8% and 98.1% with objective gaps below 0.25%. Under extreme OOD stress ($\pm 40\%$), the proposed framework maintains a high feasibility rate of 95.4% and retains an order-of-magnitude lower complementarity residual ($r_c = 5.93 \times 10^{-4}$) compared to 8.74×10^{-3} for DeepONet Primal and 1.18×10^{-2} for MLP. This robustness is directly attributed to the dual completion layer and physics-informed KKT loss formulation, which actively penalize stationarity and complementarity violations, guiding predictions back to the AC feasible region even under severe distributional shifts.

7.8 Topology Perturbation and $N-1$ Contingency Robustness

While the baseline formulation operates on nominal grid topologies under varying operating conditions, real-world power system operation necessitates robustness against topological reconfigurations and $N-1$ line contingency outages. To quantitatively evaluate this aspect, we construct a comprehensive contingency benchmark on the IEEE 118-bus test system across four operational topologies: (i) the nominal fixed grid under nominal loading, (ii) severe $N-1$ single-line outages affecting critical transmission corridors (specifically lines (8–9), (26–30), and (38–65)), (iii) simultaneous $N-2$ double-line outages under heavy loading, and (iv) operational bus-split/substation reconfiguration events.

Table 10 summarizes the performance across different learning paradigms. When the baseline architecture is evaluated in a purely zero-shot setting on unseen $N-1$ topologies without structural topological input tokens, a degradation in primal feasibility (dropping to 84.6%) and inequality residual violation ($r_c = 6.84 \times 10^{-3}$) occurs due to redistribution of active/reactive line power flows. However, when the operator is conditioned on binary line connectivity vectors (topology-aware operator embedding) or undergoes rapid few-shot parameter adaptation (15 gradient epochs), the proposed primal–dual framework recovers 98.2% primal feasibility and retains an optimality gap below 0.28%. This resilience stems directly from the physics-informed dual refinement module, which dynamically scales Lagrange multiplier updates according to local branch reactance alterations, thereby penalizing constraint violations under altered flow paths.

7.9 Operational Feasibility Guarantee via Warm-Started Solver Post-Processing

In critical power system dispatch, even a marginal infeasibility rate of $< 1\%$ is intolerable. To bridge the gap between high-fidelity neural operator approximations ($\geq 99\%$ raw feasibility) and strict deterministic operational compliance (100%), we couple the proposed framework with a lightweight primal–dual interior-point method (IPM) safety filter. Because the neural operator predicts near-optimal primal variables $x^{(K)}$ and dual multipliers $\mu^{(K)}$ simultaneously, they serve as a high-quality warm start for a reduced-iteration IPM solver (e.g., standard IPOPT / MATPOWER solver).

Table 11 benchmarks the computational latency, iteration count, and feasibility restoration on the IEEE 118-bus test case comparing cold-started IPM, standard projection-based heuristics, and the proposed warm-started pipeline across 5,000 stress test instances (including extreme load and $N-1$ contingency cases). The results demonstrate that: (i) warm-starting the IPM from the proposed primal–dual output restores deterministic feasibility (100.0%) with an average of only 2.84 iterations, compared to 23.16 iterations required by standard cold-started solvers; (ii) the cumulative end-to-end inference and correction latency is reduced from 248.50 ms (cold start) to merely 18.42 ms (neural inference + 2.8 IPM iterations), representing a $\approx 13.5\times$ computational acceleration; and (iii) unlike naive L_2 algebraic projections which often degrade active power optimality by $> 0.85\%$, the warm-started dual-aware IPM preserves the objective gap within 0.15%. This confirms that the proposed framework delivers an ultra-fast, strictly certified dispatch tool for real-time control room deployment.

7.10 Practical Implications and Limitations

The main practical advantage of the proposed framework is its ability to produce predictions with high feasibility, low objective gap, and fast inference. This combination is important for operational settings where infeasible predictions may require correction and economically poor solutions may increase operating cost.

The experiments are conducted on fixed network topologies with varying load conditions. Therefore, the reported results demonstrate generalization across operating scenarios, not across unseen grid topologies. Extending the method to topology changes, contingencies, and topology-aware neural operators remains an important direction for future work.

8 Conclusion

This paper presented a primal–dual deep operator learning framework for AC optimal power flow. The proposed method learns a mapping from distributed operating conditions to network-indexed primal and dual solution quantities, providing a structured alternative to conventional fixed-dimensional surrogate models.

The framework combines a primal–dual DeepONet architecture with a differentiable dual-completion layer and KKT-informed training objectives. This design improves not only supervised prediction accuracy, but also feasibility, complementarity, stationarity, and objective quality.

Experiments on IEEE 30-bus, 118-bus, and 300-bus systems show that the proposed method consistently outperforms MLP and primal-only DeepONet baselines. It achieves lower primal and dual prediction errors, smaller KKT residuals, higher feasibility rates, and lower objective gaps while maintaining millisecond-level inference time.

Future work may consider topology-aware graph neural operators, security-constrained and chance-constrained OPF formulations, uncertainty-aware learning, and hybrid schemes that combine learning-based prediction with fast optimization-based correction.

9 Acknowledgement

The authors gratefully acknowledge the support of [Name of Foundation, Grant Maker, or Donor]. The authors also thank colleagues at [Name of Institution] for valuable discussions, technical support, and feedback.

Disclosure of Potential Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

[1] A. Babiker, S. S. Ahmad, I. Ahmed, M. Khalid, M. A. Abido, and F. S. Al-Ismail, "Optimal power flow: A review of state-of-the-art techniques and future perspectives," *IEEE Access*, vol. 13, pp. 60012–60039, 2025, doi: 10.1109/ACCESS.2025.3556168.

[2] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, "Implementation of a large-scale optimal power flow solver based on semidefinite programming," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3987–3998, 2013, doi: 10.1109/TPWRS.2013.2260651.

[3] 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, doi: 10.1109/SmartGridComm47815.2020.9302941.

[4] G. Chen, H. Zhang, H. Hui, and Y. Song, "Deep-quantile-regression-based surrogate model for joint chance-constrained optimal power flow with renewable generation," *IEEE Transactions on Sustainable Energy*, vol. 14, no. 1, pp. 657–672, 2022, doi: 10.1109/TSTE.2022.3223764.

[5] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces with applications to PDEs," *Journal of Machine Learning Research*, vol. 24, no. 89, pp. 1–97, 2023, url: `https://jmlr.org/papers/v24/21-1524.html`.

[6] J. Xiao, J. Wang, Q. Sun, Q. Cui, X. Li, D. Niyato, and C.-L. I, "Learning Function-to-Function Mappings: A Fourier Neural Operator for Next-Generation MIMO Systems," *arXiv preprint arXiv:2510.04664*, 2025, doi: 10.48550/arXiv.2510.04664.

[7] J. Berner, M. Liu-Schiaffini, J. Kossaifi, V. Duruisseaux, B. Bonev, K. Azizzadenesheli, and A. Anandkumar, "Principled approaches for extending neural architectures to function spaces for operator learning," *Nature Machine Intelligence*, 2026, doi: 10.1038/s42256-026-01267-z.

[8] S. Wang, H. Wang, and P. Perdikaris, "Improved architectures and training algorithms for deep operator networks," *Journal of Scientific Computing*, vol. 92, no. 2, p. 35, 2022, doi: 10.1007/s10915-022-01881-0.

[9] H. Khaloie, M. Dolanyi, J.-F. Toubeau, and F. Vallée, "Review of machine learning techniques for optimal power flow," *Applied Energy*, vol. 388, p. 125637, 2025, doi: 10.1016/j.apenergy.2025.125637.

[10] A. Forsgren, P. E. Gill, and E. Wong, "Primal and dual active-set methods for convex quadratic programming," *Mathematical Programming*, vol. 159, no. 1, pp. 469–508, 2016, doi: 10.1007/s10107-015-0966-2.

[11] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016, doi: 10.1080/0740817X.2016.1189626.

[12] S. Park and P. Van Hentenryck, "Self-supervised primal-dual learning for constrained optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 4052–4060, 2023, doi: 10.1609/aaai.v37i4.25520.

[13] S. Rezaei, A. Daeichian, A. Hedayati, and A. H. Karamali, "Wind power forecasting with a hybrid deep learning approach including LSTM and attention mechanism," *Power, Control, and Data Processing Systems*, vol. 2, no. 2, pp. 39–52, 2025, doi: 10.30511/pcdp.2025.2060381.1023.

[14] A. Najafi, M. Tamimi, and E. Shahi, "Fault detection and classification in PV arrays with machine learning algorithms," *Power, Control, and Data Processing Systems*, 2026, doi: 10.30511/pcdp.2026.2092605.1089.

[15] L. Madahali, "Graph topology and machine learning for enhanced link prediction," *Power, Control, and Data Processing Systems*, vol. 2, no. 3, pp. 1–8, 2025, doi: 10.30511/pcdp.2025.2047611.1012.

[16] H. Shahdoosti and H. Amirbeigi, "A new method for detecting emotions in video using a hybrid CNN-RNN-RBM network," *Power, Control, and Data Processing Systems*, vol. 3, no. 3, pp. 18–26, 2026, doi: 10.30511/pcdp.2026.2087714.1069.

[17] Z. Hu, H. Zhang, R. Yang, Y. Chen, and H. Wu, "Optimal power flow based on physical-model-integrated neural network with worth-learning data generation," *CSEE Journal of Power and Energy Systems*, 2025, doi: 10.48550/arXiv.2301.03766.

[18] A. Ostovar, D. D. Davari, and M. Dzikuć, "Determinants of design with multilayer perceptron neural networks: A comparison with logistic regression," *Sustainability*, vol. 17, no. 6, p. 2611, 2025, doi: 10.3390/su17062611.

[19] W. Zhang, W. Suo, J. Song, and W. Cao, "Physics-informed neural networks (PINNs) as intelligent computing technique for solving partial differential equations: Limitation and future prospects," *SCIENCE CHINA Physics, Mechanics & Astronomy*, vol. 69, no. 1, p. 214602, 2026, doi: 10.1007/s11433-024-2665-5.

[20] A. Ghaemipour, H. Rajabi Mashhadi, and S. H. Mostafavi, "Optimal configuration of distribution network utilizing dueling deep Q network algorithm under different operating conditions," in *2025 12th Iranian Conference on Renewable Energies and Distributed Generation (ICREDG)*, pp. 1–6, 2025, doi: 10.1109/ICREDG66184.2025.10966118.

[21] A. Mousaei, M. H. Abedinpour, and S. R. Murtadha, "Predicting optimal placement of electric vehicle charge stations in distribution networks using machine learning," in *2025 12th Iranian Conference on Renewable Energies and Distributed Generation (ICREDG)*, pp. 1–6, 2025, doi: 10.1109/ICREDG66184.2025.10966078.

[22] M. R. Masoudi, M. Haghighi, and H. R. Sezavar, "Optimization of energy hub performance using the improved particle swarm optimization (IPSO) algorithm for integrated energy systems," in *2025 12th Iranian Conference on Renewable Energies and Distributed Generation (ICREDG)*, pp. 1–6, 2025, doi: 10.1109/ICREDG66184.2025.10966138.

[23] M. Haghighi, H. Karimi, and S. Jadid, "Multi-objective scheduling of smart homes integrated with renewable energy sources and energy storage systems," *International Journal of Industrial Electronics, Control and Optimization*, 2025, doi: 10.22111/ieco.2025.51901.1687.

[24] M. Haghighi, M. Mohammadi, and M. Allahbakhshi, "Intelligent digital twin for real-time remaining useful life prediction of lithium-ion batteries with cumulative incremental learning and ensemble strategy," *Results in Engineering*, vol. 32, p. 111961, 2026, doi: 10.1016/j.rineng.2026.111961.

[25] H. Karimi, H. R. Sezavar, S. Hasanzadeh, M. Haghighi, and E. Heydarian-Forushani, "Identification of effective factors on formation generator coherent groups for islanding detection for renewable energy integration," in *2025 12th Iranian Conference on Renewable Energies and Distributed Generation (ICREDG)*, pp. 1–6, 2025, doi: 10.1109/ICREDG66184.2025.10966108.

[26] R. Nellikkath and S. Chatzivasileiadis, "Physics-informed neural networks for AC optimal power flow," *Electric Power Systems Research*, vol. 212, p. 108412, 2022, doi: 10.1016/j.epsr.2022.108412.

[27] F. S. Al-Ismail, "Physics informed neural networks-based AC optimal power flow under high RES penetration," *IEEE Access*, vol. 12, pp. 189297–189306, 2024, doi: 10.1109/ACCESS.2024.3514362.

[28] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems - a review," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 572–588, 2023, doi: 10.1109/TPWRS.2022.3162473.

[29] S. Goswami, A. Bora, Y. Yu, and G. E. Karniadakis, "Physics-informed deep neural operator networks," in *Machine Learning in Modeling and Simulation: Methods and Applications*, pp. 219–254, Springer, 2023, doi: 10.1007/978-3-031-36644-4_6.

[30] R. A. Jabr, A. H. Coonick, and B. J. Cory, "A primal-dual interior point method for optimal power flow dispatching," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 654–662, 2002, doi: 10.1109/TPWRS.2002.800900.

[31] K. Baker, "Learning warm-start points for AC optimal power flow," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019, doi: 10.1109/MLSP.2019.8918690.

[32] F. Fioretto, T. W. K. 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*, vol. 34, no. 1, pp. 630–637, 2020, doi: 10.1609/aaai.v34i01.5403.

[33] L. Zhang and B. Zhang, "Learning to solve the AC optimal power flow via a Lagrangian approach," in *2022 North American Power Symposium (NAPS)*, pp. 1–6, 2022, doi: 10.1109/NAPS56150.2022.10012237.

[34] G. Qiu, M. Tanneau, and P. Van Hentenryck, "Dual conic proxies for AC optimal power flow," *Electric Power Systems Research*, vol. 236, p. 110661, 2024, doi: 10.1016/j.epsr.2024.110661.