PIDSOLVER:A SEQUENTIAL PHYSICS-INFORMED DIFFUSION SOLVER FOR OPTIMIZATION WITH ALGEBRAIC CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed diffusion models have emerged as a promising tool for solving large optimization problems with hard constraints. These models iteratively generate solutions, thereby improving the accuracy of end-to-end artificial neural networks (ANN) based solvers. However, for a class of optimization problems involving complex algebraic constraints, a traditional solver is required to repeatedly solve for latent variables and perform gradient estimation for ANN-based solver training, incurring prohibitively high time costs. In this work, we present a sequential Physics-informed Diffusion Solver (PIDSolver) to address this challenge. Specifically, PIDSolver couples two sequential networks: the former performs a physics-informed denoising process, while the latter handles the algebraic constraints solving process and concurrently performs gradient-based corrections to guide the former in satisfying inequality constraints. The effectiveness of the proposed method is validated in practical alternating current optimal power flow scenarios, where PIDSolver generates solutions within a second-level time scale while outperforming state-of-the-art ANN-based solvers. Our code will be released at `https://anonymous.4open.science/r/PIDSolver-D822/`.

1 INTRODUCTION

Traditional nonlinear optimization methods, such as interior-point methods, often incur prohibitive computational costs when solving large nonlinear and nonconvex optimization problems. In recent years, artificial neural networks (ANNs) based solvers, owing to their strong expressive power and fast inference speed, have gained widespread attention for optimization problems with hard constraints, such as alternating current optimal power flow (AC-OPF) in power systems. However, such problems typically involve complex algebraic constraints: during training, an additional solver, such as the Newton–Raphson method, must repeatedly solve the algebraic constraints for every batch and every sample to obtain the latent variables and complete the gradient estimation, incurring extremely high computational overhead. In particular, recent works that extend ANN-based solvers to diffusion models, which progressively restore solutions to improve solution quality, further exacerbate the challenge of training efficiency.

To address the above problem, this paper proposes the PIDSolver for optimization with complex algebraic constraints, which significantly improves solution accuracy at the cost of only a modest increase in computation time. The main contributions are summarized as follows:

- We propose a sequential physics-informed diffusion architecture that couples a multi-step noise prediction network with an algebraic constraint evaluation network, thereby incorporating the algebraic constraints into the denoising process to achieve efficient training;
- We design a physics-informed denoising loss function that jointly accounts for the optimization objectives as well as the equality and inequality constraints;
- We employ an accelerated sampling strategy that reduces the inferring time to the order of seconds;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We validate the proposed framework on the nonconvex, strongly nonlinear and large AC-OPF problem and provide an error analysis. The framework can be readily integrated into the training pipelines of existing diffusion models.

Although this paper focuses on the case of nonlinear equality constraints, the proposed framework is equally applicable to other differential equality constraints (whose objectives can likewise be realized by physics-informed neural networks).

2 BACKGROUND

2.1 AC-OPF PROBLEM STATEMENT

The AC-OPF is mathematically formulated as a nonlinear optimization problem. It aims to minimize the total generation cost $\min \sum_{i\in\mathcal{G}} C_i(P_{Gi})$ while satisfying the physical laws of the power grid and operational security constraints, which are defined as follows (Huang et al., 2024):

$$\begin{aligned}
&P_{Gi} - P_{Di} = V_i \sum_{j\in\mathcal{N}} V_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}), \forall i \in \mathcal{N} \\
&Q_{Gi} - Q_{Di} = V_i \sum_{j\in\mathcal{N}} V_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}), \forall i \in \mathcal{N} \\
&P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max}, \quad Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}, \forall i \in \mathcal{G} \\
&V_i^{\min} \le V_i \le V_i^{\max}, \forall i \in \mathcal{N}, \quad |S_{ij}| \le S_{ij}^{\max}, \forall (i,j) \in \mathcal{L}
\end{aligned} \tag{1}$$

where $C_i(\cdot)$ is the cost function of generator i, P_{Gi} and Q_{Gi} represent the active and reactive power of generators, P_{Di} and Q_{Di} denote the active and reactive load, V_i and θ_i are the voltage magnitude and phase angle at bus i, and G_{ij} and B_{ij} are the elements of the admittance matrix $\mathbf{Y}$. The sets $\mathcal{N}$, $\mathcal{G}$, and $\mathcal{L}$ refer to the collections of all buses, generators, and branches, respectively.

Let $\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z})$ denote the AC-PF equality constraints in equation 1, where $\mathbf{x}_0$ represents the AC-OPF solution, $\mathbf{y}$ includes the load demands and the network admittance matrix, and $\mathbf{z}$ represents the remaining variables derived from the alternating current power flow (AC-PF) calculation. According to the standard classification of buses in power system analysis, namely, PQ, PV, and slack ($V\delta$) buses (Donti et al., 2021), the variables are partitioned as

$$\begin{aligned}
&\mathbf{x}_0 \triangleq (\mathbf{P}_{\mathrm{PV}}, \mathbf{U}_{\mathrm{PV}}, U_{V\delta}, \delta_{V\delta}), \mathbf{y} \triangleq (\mathbf{P}_{\mathrm{PQ}}, \mathbf{Q}_{\mathrm{PQ}}, \mathbf{Y}), \\
&\mathbf{z} \triangleq (\mathbf{U}_{\mathrm{PQ}}, \delta_{\mathrm{PQ}}, \delta_{\mathrm{PV}}, P_{V\delta}, \mathbf{Q}_{\mathrm{PV}}, Q_{V\delta}).
\end{aligned} \tag{2}$$

With these notations, the AC-OPF problem in equation 1 can be compactly written as

$$\min_{\mathbf{x}_0} \; f(\mathbf{x}_0) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}) = \mathbf{0}, \;\; \mathbf{g}(\mathbf{x}_0, \mathbf{z}) \le \mathbf{0}, \tag{3}$$

where $f(\mathbf{x}_0) = \sum_{i\in\mathcal{G}} C_i(P_{Gi})$ denotes the total generation cost, $\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z})$ denotes the AC-PF equality constraints, and $\mathbf{g}(\mathbf{x}_0, \mathbf{z})$ collects the inequality constraints in equation 1.

2.2 RELATED WORK

As shown in problem equation 3, AC-OPF is nonconvex, nonlinear, large, and constraint-heavy. Traditional nonlinear optimization solvers, such as interior-point methods (Wächter & Biegler, 2006; Merkli et al., 2018) and sequential quadratic programming (Hasan et al., 2024), are reliable but become time-consuming as the system size and number of constraints grow. To improve computational speed, AC-OPF is often addressed using approximations and/or relaxation techniques (Hasan et al., 2025). However, this may come at the cost of model accuracy: the resulting solution may be suboptimal or may even violate constraints under the AC-PF used for validation, rendering the solution infeasible and thereby compromising power system security and economic performance.

In recent years, end-to-end ANN based AC-OPF solving methods have attracted significant research interest (Babiker et al., 2025; Mohammadi et al., 2024). These methods learn the end-to-end mapping from load profiles and admittance matrix $\mathbf{y}$ to AC-OPF solutions $\mathbf{x}_0$ and, once sufficiently

trained, can directly produce optimized AC-OPF results within milliseconds under given load conditions (Pan et al., 2019; Huang et al., 2022; Baker, 2022; Pan et al., 2023). Their training typically adopts a supervised regression loss that drives the predicted solution toward a reference optimal solution:

$$\mathcal{L}_{\text{ANN}}(\boldsymbol{\Theta}) = \mathbb{E}_{\mathbf{y}}\left[\|\hat{\mathbf{x}}_0(\mathbf{y};\boldsymbol{\Theta}) - \mathbf{x}_0^*(\mathbf{y})\|_2^2\right], \tag{4}$$

where $\hat{\mathbf{x}}_0(\mathbf{y};\boldsymbol{\Theta})$ denotes the AC-OPF solution predicted by the network and $\mathbf{x}_0^*(\mathbf{y})$ the reference optimal solution.

Physics-informed Neural Network To further enhance generalization ability and physical consistency, several studies (Donti et al., 2021; Chen et al., 2025; Huang et al., 2024) employed physics-informed neural networks and directly embed physical laws and operational constraints into the learning process by augmenting the above loss with a constraint-violation penalty term:

$$\mathcal{L}(\boldsymbol{\Theta}) = \mathcal{L}_{\text{ANN}}(\boldsymbol{\Theta}) + \lambda\,\mathbb{E}_{\mathbf{y}}\Big[\big\|\boldsymbol{\mathcal{R}}_{\mathbf{g}}\big(\hat{\mathbf{x}}_0(\mathbf{y};\boldsymbol{\Theta}),\mathbf{z}\big)\big\|_2^2\Big], \tag{5}$$

where $\lambda > 0$ is a penalty coefficient and $\boldsymbol{\mathcal{R}}_{\mathbf{g}}(\hat{\mathbf{x}}_0,\mathbf{z}) = \max\big(\mathbf{g}(\hat{\mathbf{x}}_0,\mathbf{z}),\mathbf{0}\big)$ denotes the inequality constraint violation. Once the networks are well trained, they can infer the solution in real time.

In intra-day rolling dispatch scenarios, many electricity markets adopt a 5-minute decision cycle (15 minutes in some regions) (IEA, 2025) and perform rolling optimization to continuously update operational plans. This observation raises a key question: can a modest increase in computation time (from milliseconds to seconds) be traded for more accurate and more reliable AC-OPF solutions?

Denoising Diffusion Probabilistic Model To this end, denoising diffusion probabilistic model (DDPM) based solver for the AC-OPF problem is proposed (Hoseinpour & Dvorkin, 2026), which decomposes the solution process into a forward diffusion stage that progressively injects noise and a backward generation stage that progressively denoises. Building on this work, a physics-guided correction step into the backward denoising process is further developed to improve solution feasibility (Shekhar et al., 2026). The training objective of these methods is the standard diffusion model denoising loss (Ho et al., 2020):

$$\mathcal{L}_{\text{DDPM}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{y},t)\|_2^2\right], \quad \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{6}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$ is the cumulative product of the noise schedule $\{\alpha_s\}_{s=1}^{T}$. The physics-guided correction step is equivalent to applying a gradient correction to the reconstructed solution during backward denoising, i.e., $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_0 - \gamma\nabla_{\hat{\mathbf{x}}_0}\mathcal{V}(\hat{\mathbf{x}}_0)$, where $\mathcal{V}(\cdot)$ measures the constraint-violation level of a solution. Nevertheless, none of the above methods directly embed physical constraints into the DDPM training process. Consequently, the physical consistency of the generated solutions remains difficult to guarantee.

Physics-informed Diffusion Models Physics-informed DDPM (PIDDPM) introduce physical constraints as guidance signals during the denoising process (Bastek et al., 2025). This progressively restore solutions that obey the physical laws during denoising and markedly improving solution quality and physical consistency. The objective function of such methods typically augments the denoising loss with a physics-informed term:

$$\mathcal{L}_{\text{PIDDPM}}(\boldsymbol{\Theta}) = \mathbb{E}_{\mathbf{x}_0,t,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}}(\mathbf{x}_t,\mathbf{y},t)\|_2^2 + \lambda\left\|\boldsymbol{\mathcal{R}}_{\mathbf{g}}\big(\hat{\mathbf{x}}_0(\mathbf{x}_t),\mathbf{z}\big)\right\|_2^2\right], \tag{7}$$

where $\boldsymbol{\mathcal{R}}(\hat{\mathbf{x}}_0)$ denotes the constraint violation of the generated solution.

Notably, for both physics-informed end-to-end and diffusion solvers, evaluating the residual $\boldsymbol{\mathcal{R}}_{\mathbf{g}}\big(\hat{\mathbf{x}}_0,\mathbf{z}\big)$ relies on the latent variable $\mathbf{z}$, which must be obtained by solving the AC-PF equality constraints $\mathbf{h}(\hat{\mathbf{x}}_0,\mathbf{y},\mathbf{z}) = \mathbf{0}$. Consequently, an additional solver (generally the Newton–Raphson method (da Costa et al., 1999)) must be repeatedly invoked for every batch and every sample to solve the algebraic constraints for the latent variables. Gradient estimation must be performed along this implicit solution path, which incurs extremely high computational costs.

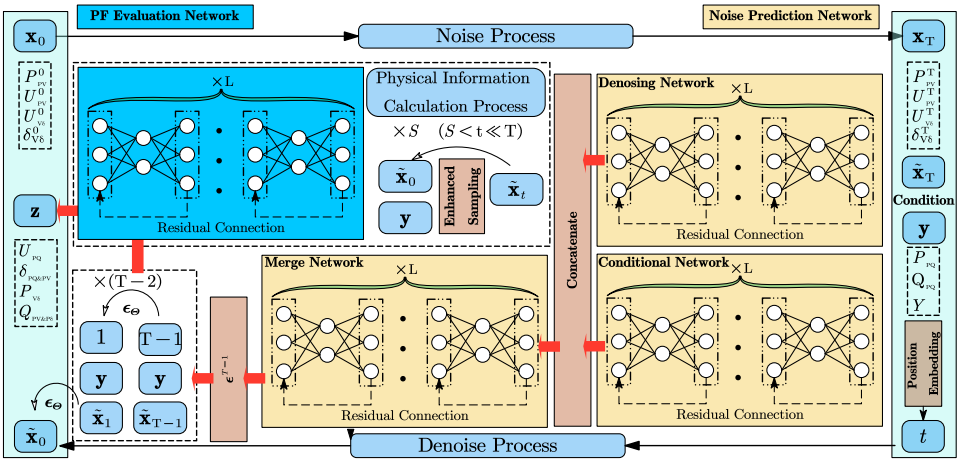

Figure 1: The Detailed Model Structure.

3 Sequential Physics-informed Diffusion Solver for Optimization with Algebraic Constraints

3.1 The Sequential Architecture Design

We adopt the DDPM formulation reviewed in Section 2.2 and an approximate clean solution is reconstructed in closed form:

$$\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_{\boldsymbol{\Theta}}(\mathbf{x}_t, \mathbf{y}, t)}{\sqrt{\bar{\alpha}_t}}. \tag{8}$$

As there is no gradient flow between the constraints $\mathbf{g}$ and the distribution $p_{\boldsymbol{\Theta}}(\mathbf{x}_0|\mathbf{y})$, the parameters $\boldsymbol{\Theta}$ cannot be optimized by minimizing the constraint violation $\mathbf{g}$. Therefore, we propose a sequential network shown in Fig. 1: a noise prediction network (parameterized as $\boldsymbol{\Theta}_1$) predicts the noise $\boldsymbol{\epsilon}_t$ from $(\mathbf{x}_t, \mathbf{y}, t)$ to compute $\mathbf{x}_0$ via equation 8, which is then fed with $\mathbf{y}$ into an AC-PF evaluation network (parameterized as $\boldsymbol{\Theta}_2$) to obtain $\mathbf{z}$ and the residual $\boldsymbol{\mathcal{R}}(\mathbf{x}_0) = \begin{bmatrix} \mathbf{g}(\mathbf{x}_0, \mathbf{z}) \\ \mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}) \end{bmatrix}$. This creates a direct gradient pathway from the physics-informed loss to $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2\}$, enabling loss design and gradient propagation.

3.2 Physical-Informed Loss Function Design

We adopt an energy-based method (EBM) view to incorporate AC-OPF physics into the diffusion model. Defining the energy function $E(\mathbf{x}_0) = \frac{1}{2\sigma^2}\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0)\right\|^2$, which measures how much a sample $\mathbf{x}_0$ violates the physical laws. In an EBM framework, lower energy corresponds to higher probability $p(\mathbf{x}_0) \propto \exp\left(-E(\mathbf{x}_0)\right) = \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0)\|^2\right)$. Following the prior work (Bastek et al., 2025), we introduce the virtual observation $\hat{\mathbf{r}} = \mathbf{0}$ and assume it is generated from $q_{\boldsymbol{\mathcal{R}}}(\hat{\mathbf{r}} \mid \mathbf{x}_0) = \mathcal{N}\left(\hat{\mathbf{r}}; \boldsymbol{\mathcal{R}}(\mathbf{x}_0), \sigma^2\mathbf{I}\right)$. As $\sigma^2 \to 0$, the probability mass concentrates on $\boldsymbol{\mathcal{R}}(\mathbf{x}_0)$, enforcing the residual equation. The corresponding virtual likelihood is $p_{\boldsymbol{\Theta}}(\hat{\mathbf{r}}) = \int q_{\boldsymbol{\mathcal{R}}}(\hat{\mathbf{r}} \mid \mathbf{x}_0)\, p_{\boldsymbol{\Theta}}(\mathbf{x}_0)\, d\mathbf{x}_0 = \mathbb{E}_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left[q_{\boldsymbol{\mathcal{R}}}(\hat{\mathbf{r}} \mid \mathbf{x}_0)\right]$. We maximize the log-likelihood of the virtual observation:

$$\arg\max_{\boldsymbol{\Theta}} \; \mathbb{E}_{\hat{\mathbf{r}}}\left[\log p_{\boldsymbol{\Theta}}(\hat{\mathbf{r}})\right] = \arg\max_{\boldsymbol{\Theta}} \; \mathbb{E}_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left[\log q_{\boldsymbol{\mathcal{R}}}(\hat{\mathbf{r}} = \mathbf{0} \mid \mathbf{x}_0)\right]. \tag{9}$$

Since $\log q_{\boldsymbol{\mathcal{R}}}(\hat{\mathbf{r}} = \mathbf{0} \mid \mathbf{x}_0) = -\frac{1}{2\sigma^2}\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0)\right\|^2 + \text{const}$, maximizing the virtual log-likelihood above is equivalent to minimizing the expected squared residual $\mathbb{E}_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left[\frac{1}{2\sigma^2}\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0)\right\|^2\right]$ (Bastek et al., 2025). Finally, we obtain the PIDSolver training objective:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \mathbb{E}_{\mathbf{x}_0, t, \boldsymbol{\epsilon}}\Big[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}(\mathbf{x}_t, \mathbf{y}, t)\right\|^2 + \lambda_h\left\|\mathbf{h}(\mathbf{x}_0^*, \mathbf{y}, \mathbf{z}^*)\right\|^2 + \lambda_g\left\|\mathbf{g}(\mathbf{x}_0^*, \mathbf{z}^*)\right\|^2 + \lambda_c\, C(\mathbf{x}_0^*)\Big], \tag{10}$$

where $\mathbf{x}_0^* = \mathbf{x}_0^*(\mathbf{x}_t; \boldsymbol{\Theta}_1)$ denotes the approximate clean solution reconstructed by the noise prediction network from the noisy state $\mathbf{x}_t$ (using an enhanced sampling method described in the next subsection), $\mathbf{z}^* = \mathbf{z}_{\boldsymbol{\Theta}_2}(\mathbf{x}_0^*, \mathbf{y})$ denotes the latent variable produced by the AC-PF evaluation network, C denotes the quadratic generation cost of the reconstructed dispatch, normalized by the reference value 10^5 \$/h, and λ_h, λ_g, λ_c are regularization coefficients weighting the equality residual, the inequality residual, and the operating cost, respectively.

Two remarks distinguish equation 10 from the generic physics-informed losses reviewed in Section 2.2. First, the objective depends explicitly on the parameters of both networks, $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$, and the combined residual is decoupled into an equality residual $\mathbf{h}$ and an inequality residual $\mathbf{g}$ with separate coefficients. Second, and more importantly, the latent variable $\mathbf{z}^*$ is generated in closed form by the AC-PF evaluation network instead of being recovered by repeatedly invoking an external AC-PF solver, which renders equation 10 fully differentiable end-to-end. The gradients of the two residual terms propagate along the two network paths as follows: the equality-residual gradient flows to $\boldsymbol{\Theta}_2$ through $\partial\mathbf{h}/\partial\mathbf{z}\cdot\partial\mathbf{z}^*/\partial\boldsymbol{\Theta}_2$, and to $\boldsymbol{\Theta}_1$ through $\mathbf{x}_0^*$ both directly ($\partial\mathbf{h}/\partial\mathbf{x}_0$) and indirectly via the AC-PF evaluation network ($\partial\mathbf{h}/\partial\mathbf{z}\cdot\partial\mathbf{z}^*/\partial\mathbf{x}_0^*$); the inequality-residual gradient follows the same two paths, with the important difference that the violations of the inequality constraints can only be evaluated once the latent variables are available, so a portion of its gradient necessarily traverses the AC-PF evaluation network. Applying the chain rule to the physics-informed term $\mathcal{L}_{\text{phy}}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \lambda_h \big\|\mathbf{h}(\mathbf{x}_0^*, \mathbf{y}, \mathbf{z}^*)\big\|_2^2 + \lambda_g \big\|\mathbf{g}(\mathbf{x}_0^*, \mathbf{z}^*)\big\|_2^2 + \lambda_c\, C(\mathbf{x}_0^*)$ yields the explicit decompositions

$$\nabla_{\boldsymbol{\Theta}_1}\mathcal{L}_{\text{phy}} = \mathbb{E}\Big[2\Big(\lambda_h \mathbf{h}^\top \frac{\partial \mathbf{h}}{\partial \mathbf{x}_0} + \lambda_g \mathbf{g}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{x}_0} + \Big(\lambda_h \mathbf{h}^\top \frac{\partial \mathbf{h}}{\partial \mathbf{z}} + \lambda_g \mathbf{g}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{z}}\Big)\frac{\partial \mathbf{z}^*}{\partial \mathbf{x}_0^*}\Big)\frac{\partial \mathbf{x}_0^*}{\partial \boldsymbol{\Theta}_1} + \lambda_c \frac{\partial C}{\partial \mathbf{x}_0^*}\frac{\partial \mathbf{x}_0^*}{\partial \boldsymbol{\Theta}_1}\Big], \tag{11}$$

$$\nabla_{\boldsymbol{\Theta}_2}\mathcal{L}_{\text{phy}} = \mathbb{E}\Big[2\Big(\lambda_h \mathbf{h}^\top \frac{\partial \mathbf{h}}{\partial \mathbf{z}} + \lambda_g \mathbf{g}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{z}}\Big)\frac{\partial \mathbf{z}^*}{\partial \boldsymbol{\Theta}_2}\Big], \tag{12}$$

where all partial derivatives are evaluated at $(\mathbf{x}_0^*, \mathbf{y}, \mathbf{z}^*)$. Notably, the inequality-residual gradient w.r.t. $\boldsymbol{\Theta}_2$ propagates exclusively through the AC-PF evaluation network via $\partial\mathbf{z}^*/\partial\boldsymbol{\Theta}_2$, and neither gradient requires implicit differentiation through the AC-PF equations; both are obtained by a single forward pass of each network followed by backpropagation. The operating-cost term $C(\mathbf{x}_0^*)$ depends only on the generated active-power outputs contained in $\mathbf{x}_0^*$: it contributes the direct gradient $\lambda_c\, \partial C/\partial\mathbf{x}_0^* \cdot \partial\mathbf{x}_0^*/\partial\boldsymbol{\Theta}_1$ in equation 11 and does not back-propagate into $\boldsymbol{\Theta}_2$.

Assumption 1 *For every candidate solution $\mathbf{x}_0$ and operating condition $\mathbf{y}$ in the operating region covered by the training distribution—i.e., load profiles obtained by perturbing each bus load independently with 10% standard deviation around the nominal profile—the AC-PF equations admit a unique implicit solution $\mathbf{z}^\dagger(\mathbf{x}_0, \mathbf{y})$ with $\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}^\dagger) = \mathbf{0}$, $\mathbf{h}$ and $\mathbf{g}$ are L_h- and L_g-Lipschitz continuous in $\mathbf{z}$, respectively, and the residual map $\boldsymbol{\mathcal{R}}$ is L_x-Lipschitz continuous in $\mathbf{x}_0$.*

Lemma 1 (Surrogate fidelity of the AC-PF evaluation network) *Let $\hat{\mathbf{z}} = \mathbf{z}_{\boldsymbol{\Theta}_2}(\mathbf{x}_0, \mathbf{y})$ be the surrogate produced by the AC-PF evaluation network with error $\big\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\big\|_2 \le \epsilon_z$. Under Assumption 1, the residual evaluated with the surrogate deviates from its exact counterpart by*

$$\Big\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}}) - \boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\Big\|_2 \le \sqrt{L_h^2 + L_g^2}\,\epsilon_z,$$

and, in particular, the equality residual computed with the surrogate satisfies $\big\|\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \hat{\mathbf{z}})\big\|_2 \le L_h\epsilon_z$. Hence the physics-informed term in equation 10 evaluated with the ANN surrogate is an $O(\epsilon_z)$-accurate approximation of the term one would compute with an exact AC-PF solver. Moreover, the pre-training objective $\mathcal{L}(\boldsymbol{\Theta}_2) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}\big[\|\hat{\mathbf{z}} - \mathbf{z}\|_2^2 + \eta\|\boldsymbol{\mathcal{R}}_{\mathbf{h}}(\mathbf{x}, \mathbf{y}, \hat{\mathbf{z}})\|_2^2\big]$ drives the expected surrogate error ϵ_z toward zero, so the surrogate fidelity is directly controlled by the pre-training quality of the AC-PF evaluation network.

Proof 1 *The proof can be found in Appendix A.1.*

Theorem 1 (Probabilistic feasibility of the generated solutions) *Let $p_{\boldsymbol{\Theta}}$ denote the generative model parameterized by $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2\}$, and suppose that the residual term of the physics-informed objective in equation 10 is optimized such that $\mathbb{E}_{\mathbf{x}_0\sim p_{\boldsymbol{\Theta}}}\big[\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}})\|_2^2\big] \le \delta^2$ for some $\delta \ge 0$. Under*

Assumption 1 with surrogate error ϵ_z, for any tolerance $\varepsilon > 0$ the true residual of the generated solutions satisfies

$$P_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left(\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2 > \varepsilon\right) \leq \frac{\delta + \sqrt{L_h^2 + L_g^2}\,\epsilon_z}{\varepsilon},$$

and $\mathbb{E}_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left[\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\|_2\right] \leq \delta + \sqrt{L_h^2 + L_g^2}\,\epsilon_z$. *In particular, as $\delta \to 0$ and $\epsilon_z \to 0$, the generative distribution concentrates on the true feasible set $\{\mathbf{x}_0 : \boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger) = \mathbf{0}\}$ in probability.*

Proof 2 *The proof can be found in Appendix A.2.*

In the following 118-bus experiments (Section 4), the trained PIDSolver attains an expected residual $\mathbb{E}\left[\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}})\|_2^2\right] \approx 1.1 \times 10^{-2}$ on the held-out validation set (i.e., $\delta \approx 0.10$ in Theorem 1), and the AC-PF evaluation network reaches a validation MSE of $\approx 2 \times 10^{-4}$ (i.e., $\epsilon_z \approx 1.4 \times 10^{-2}$), which numerically instantiates the two error sources in the probabilistic feasibility bound.

3.3 Enhanced Sampling Strategy

During training, evaluating the physics-informed residual $\boldsymbol{\mathcal{R}}(\mathbf{x}_0)$ requires a clean estimate of $\mathbf{x}_0$ from a noisy state $\mathbf{x}_t$. While a full T-step reverse diffusion process is computationally prohibitive, a simple one-shot reconstruction often lacks the accuracy needed to provide reliable gradients for physics-based loss functions. We therefore adopt an accelerated multi-step sampling strategy (Song et al., 2021; Bastek et al., 2025): starting from the noisy state $\mathbf{x}_t$ at an arbitrary diffusion step t, we define a subsequence of S time steps $\{\tau_i\}_{i=0}^{S}$ with $\tau_S = t$, $\tau_0 = 0$, and $S \ll T$. For $i = S, S-1, \ldots, 1$, the state is updated deterministically as

$$\mathbf{x}_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}}\left(\frac{\mathbf{x}_{\tau_i} - \sqrt{1-\bar{\alpha}_{\tau_i}}\boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}(\mathbf{x}_{\tau_i}, \mathbf{y}, \tau_i)}{\sqrt{\bar{\alpha}_{\tau_i}}}\right) + \sqrt{1-\bar{\alpha}_{\tau_{i-1}}}\boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}(\mathbf{x}_{\tau_i}, \mathbf{y}, \tau_i), \tag{13}$$

with $\bar{\alpha}_{\tau_0} = 1$, so that the last step returns the clean solution. The final state $\mathbf{x}_{\tau_0}$ is taken as the approximate clean solution $\mathbf{x}_0^*(\mathbf{x}_t; \boldsymbol{\Theta}_1)$ fed to the physics-informed terms in equation 10, i.e.,

$$\mathbf{x}_0^*(\mathbf{x}_t; \boldsymbol{\Theta}_1) = \mathbf{x}_{\tau_0} = \big(\mathcal{D}_1 \circ \mathcal{D}_2 \circ \cdots \circ \mathcal{D}_S\big)(\mathbf{x}_t), \tag{14}$$

where $\mathcal{D}_i(\cdot)$ denotes the deterministic transition in equation 13 from τ_i to τ_{i-1}.

The necessity of the multi-step refinement follows from the standard relation between the one-shot reconstruction error and the noise prediction error of DDPM (Ho et al., 2020; Song et al., 2021): substituting equation 6 into equation 8 yields $\mathbb{E}\left[\left\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\right\|_2^2\right] = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\,\mathbb{E}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}(\mathbf{x}_t, \mathbf{y}, t)\right\|_2^2\right] \leq \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}\,\epsilon_{\text{pred}}^2$, where ϵ_{pred}^2 upper-bounds the denoising loss. Since the amplification factor $(1-\bar{\alpha}_t)/\bar{\alpha}_t$ diverges as t approaches T, a single reconstruction step is unreliable for physics evaluation at early diffusion steps. The S-step refinement in equation 13 and equation 14 progressively reduces this error toward zero as the trajectory approaches τ_0, which is corroborated empirically in Table 2. By Assumption 1, the residual evaluation error of any reconstructed solution decomposes as

$$\left\|\boldsymbol{\mathcal{R}}\big(\hat{\mathbf{x}}_0, \hat{\mathbf{z}}\big) - \boldsymbol{\mathcal{R}}\big(\mathbf{x}_0, \mathbf{z}^\dagger\big)\right\|_2 \leq L_x\left\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\right\|_2 + \sqrt{L_h^2 + L_g^2}\,\epsilon_z, \tag{15}$$

where the decomposition follows from the triangle inequality, the first term is bounded by the L_x-Lipschitz continuity of $\boldsymbol{\mathcal{R}}$ in $\mathbf{x}_0$ (Assumption 1), and the second term is bounded by Lemma 1. Hence the physics-informed gradients in equation 11 and equation 12 remain faithful whenever both the denoising error and the surrogate error are small. Together with Theorem 1, this closes the error chain from the training objectives to the feasibility of the generated solutions.

Because each transition $\mathcal{D}_i$ is a differentiable function of $\boldsymbol{\Theta}_1$, the reconstruction map $\mathbf{x}_0^*(\mathbf{x}_t; \boldsymbol{\Theta}_1)$ is differentiable in $\boldsymbol{\Theta}_1$, and the gradient $\partial \mathbf{x}_0^* / \partial \boldsymbol{\Theta}_1$ appearing in equation 11 is computed by unrolling the S-step recursion in equation 13 and equation 14 via backpropagation. In this way, the sampled solution participates directly in the physics-informed training: the terms $\left\|\mathbf{h}(\mathbf{x}_0^*, \mathbf{y}, \mathbf{z}^*)\right\|^2$ and $\left\|\mathbf{g}(\mathbf{x}_0^*, \mathbf{z}^*)\right\|^2$ in equation 10 penalize the constraint violations of the reconstructed solution $\mathbf{x}_0^*$, and the penalty gradients propagate backward along the S sampling steps into the noise prediction network, steering the entire denoising trajectory toward physically feasible solutions. Compared with

a full T-step reverse process, this strategy reduces the reconstruction cost from T to $S \ll T$ network evaluations per sample while retaining sufficient reconstruction accuracy for reliable physics gradients. At inference, the same deterministic update is applied along the full trajectory from $\tau_S = T$ down to $\tau_0 = 0$. From an EBM perspective, the physics-informed term in equation 10 steers each denoising step toward a descent direction of the energy $E(\mathbf{x}_0) = \frac{1}{2\sigma^2}\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0)\right\|^2$ on the reconstructed clean solutions: as t approaches zero, the denoising direction of the distribution $p_{\boldsymbol{\Theta}}(\mathbf{x}_0) \propto \exp\big(-E(\mathbf{x}_0)\big)$ increasingly coincides with $-\nabla E$, so the residual of the reconstructed solutions is expected to decrease monotonically along the sampling path. This expectation is consistent with the empirical results in Table 2, where the constraint violations drop by more than an order of magnitude as the number of inference steps increases from 5 to 100.

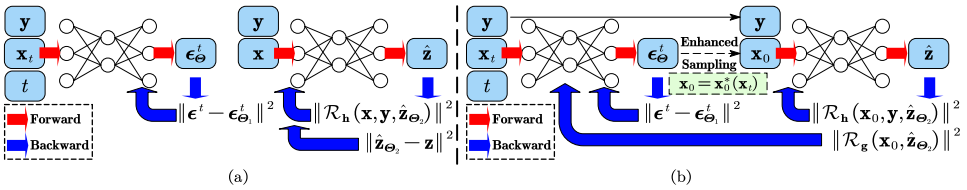

Figure 2: Overall Training Process ((a) Pre-training Stage, (b) Fine-tuning Stage).

3.4 The Overall Training Process

The overall training process consists of two stages: pre-training and fine-tuning, as shown in Fig 2. During pre-training in Fig 2 (a), the noise prediction network takes $(\mathbf{x}_t, \mathbf{y}, t)$ as input, predicts the noise $\boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}(\mathbf{x}_t, \mathbf{y}, t)$, and is pre-trained with the DDPM loss in equation 6. The AC-PF evaluation network is pre-trained with a combination of a mean squared error loss and a AC-PF imbalance loss, i.e., $\mathcal{L}(\boldsymbol{\Theta}_2) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}\left[\|\hat{\mathbf{z}} - \mathbf{z}\|^2 + \lambda_h \|\mathbf{h}\|^2\right]$, where $\hat{\mathbf{z}} = \mathbf{z}_{\boldsymbol{\Theta}_2}(\mathbf{x}, \mathbf{y})$. The fine-tuning process is shown in Fig. 2 (b). In the forward process, PIDSolver takes condition $\mathbf{y}$, diffusion step t, and noisy state $\mathbf{x}_t$. It outputs predicted noise $\boldsymbol{\epsilon}^t_{\boldsymbol{\Theta}_1}(\mathbf{x}_t, \mathbf{y}, t)$. In the backward process, parameters are updated by minimizing the noise discrepancy in equation 10. Meanwhile, AC-PF evaluation network calculates AC-PF residuals and inequality constraint violations $\boldsymbol{\mathcal{R}}(\mathbf{x}_0)$ to enforce AC-OPF consistency. During the joint fine-tuning of equation 10, the constraint-residual term $\lambda_h\|\mathbf{h}\|^2 + \lambda_g\|\mathbf{g}\|^2$ is monitored every two epochs on the held-out validation set, and training stops early if it fails to improve for ten consecutive evaluations.

4 Case Study

4.1 Illustrative Examples

To build intuition for the proposed framework before the large AC-OPF experiments, we illustrate its core ingredients on three low-dimensional examples built on the objective $f(x, y) = (x-2)^2 + (y-3)^2$, in which the diffusion process operates in a three-dimensional state space. The corresponding trajectory visualizations are collected in Appendix B. Across the three cases, a consistent trend emerges: denoising trajectories whose starting points lie outside the feasible region undergo a marked shift toward the feasible region as denoising progresses, whereas trajectories starting inside the feasible region follow nearly straight paths.

4.2 Data Set

This paper employs PowerModel.jl (Coffrin et al., 2018) to generate AC-OPF data and evaluates PIDSolver on the standard IEEE 118-bus system, which is larger and more complex than the IEEE 57-bus system tested in (Donti et al., 2021). By perturbing each bus load independently around its nominal value with 10% standard deviation, as in (Pan et al., 2023), 12,000 sample points are generated. The dataset is partitioned into three subsets (10:1:1): a training set of 10,000 samples, a validation set of 1,000 samples (used for early stopping of the fine-tuning stage) and a test set of 1,000 samples. The loads and the generator active-power outputs are normalized by the 100 MVA system

base, while voltages, reactive-power outputs and voltage angles are already in per-unit form; no additional normalization is applied in this study. Following evaluation metrics are considered (Chen et al., 2025; Huang et al., 2024): (1) the operating cost gap $l_{cost} = \frac{C-C_o}{C_o} \times 100\%$, where C and C_o denote the generation cost given by ANN and PowerModel.jl, respectively; (2) average violation $\bar{l}_v = \frac{54\,l_p+54\,l_q+118\,l_u+186\,l_{line}}{412}$, which weights each constraint type by the number of corresponding system components (54 generator active/reactive limits, 118 bus voltage limits, and 186 branch flow limits); (3) individual constraint violations including active power limits l_p, reactive power limits l_q, bus voltage limits l_u and line-flow limits l_{line}.

4.3 Results Analysis

Performance Comparison A comprehensive performance comparison is presented in Table 1, incorporating multiple state-of-the-art methods to ensure rigorous assessment. The supervised method (Huang et al., 2022) (denoted as M1), the latest PINN-based method (Chen et al., 2025) (denoted as M2), the DC3-based method (Donti et al., 2021) (denoted as M3), the DDPM-based method (Hoseinpour & Dvorkin, 2026) (denoted as M4) and the fast DDPM-based method (denoted as M5) (Shekhar et al., 2026) are taken into comparison. The proposed method structure and training parameters are detailed in Table C.3 of Appendix C. It is worth noting that the PIDSolver results correspond to a diffusion step count of $T = 1000$. The results in Table 1 demonstrate that the proposed PIDSolver achieves the lowest constraint violations among the learning-based methods, reducing the average violation by more than one orders of magnitude compared to M2. The cost obtained by PIDSolver is very close to that of PowerModel.jl. This indicates that PIDSolver effectively balances solution accuracy and physical feasibility, validating the benefits of the stepwise physical-informed diffusion approach. It is worth noting that M3 results are absent, as the method requires Jacobian inversion. AC-PF is strongly nonlinear, and this inversion causes severe numerical instability in large systems; similar results are presented in (Huang et al., 2024).

Table 1: Comprehensive Performance Comparison.

Methods	l_{cost} (%)	Constraint Violations (10^{-2})				Time
		$\bar{l}_v$	l_p	l_q	l_u	
M1	0.12	1.38	0.07	10.25	0.08	$\leq$ 0.001s
M2	0.07	0.83	0.02	8.46	0.01	$\leq$ 0.001s
M3	-	-	-	-	-	-
M4	0.01	0.16	0.02	1.14	0.00	$\leq$ 2s
M5	0.01	0.14	0.02	1.06	0.00	$\leq$ 1s
PIDSolver	**0.01**	**0.01**	**0.00**	**0.10**	**0.00**	$\leq$ 2s

It is noteworthy that M4 and M5 achieve l_{cost}, l_p, and l_v comparable to those of PIDSolver, whereas their reactive power violations, l_q, are more than an order of magnitude larger. Reactive power limit violations can have severe consequences for power-system operation. Exceeding a generator's reactive power limits may push it beyond its excitation capability, potentially causing overheating, triggering protection mechanisms, and depleting the reactive power reserves needed to maintain voltage stability. Such violations are also difficult to mitigate in ANN-based solvers because reactive power flows depend strongly and nonlinearly on voltage magnitudes through the AC-PF equations. Consequently, even small errors in AC-OPF solutions can lead to non-negligible reactive power imbalances. Moreover, l_q is in per-unit values normalized by the system base capacity. Thus, l_q must be multiplied by the 100 MVA base to obtain the corresponding physical imbalance. For example, the violations produced by M4 and M5 correspond to approximately 1 MVA. PIDSolver provides a great improvement in this safety-critical metric, which is important for secure power system operation.

Inferring Time and Accuracy Trade-off Discussion To further explore how the number of inference steps affects the performance of PIDSolver, we present a trade-off analysis in Table 2, where all timings are represent actual execution times for a 1,000-sample batch on one RTX 5090 GPU. The accelerated S-step strategy reduces inferring time linearly in S, at 0.19 s for S=100 and 0.02 s for S=5. Quality and latency can therefore be traded over one orders of magnitude, and the S=20–100 regime keeps $l_q \leq 1.02 \times 10^{-2}$ at 0.05–0.19 s per 1,000 samples. Furthermore, since PIDSolver generates solutions iteratively, it does not increase inference

Table 2: Inference Steps, Time and Accuracy Trade-off of PIDSolver.

Steps S	Time/s.	l_{cost} (%)	Constraint Violations (10^{-2})			
			$\bar{l}_v$	l_p	l_q	l_u
5	0.018	0.07	0.89	0.00	6.80	0.00
10	0.027	0.02	0.32	0.00	2.45	0.00
20	0.046	0.01	0.13	0.00	1.02	0.00
50	0.100	0.01	0.06	0.00	0.42	0.00
100	0.194	0.01	0.04	0.00	0.27	0.00
1000	1.672	0.01	**0.01**	**0.00**	**0.10**	**0.00**

memory usage compared to existing end-to-end ANN-based solvers.

Training Efficiency The results in Table 3 further demonstrate the training efficiency of PIDSolver. All solvers are trained and evaluated on a single NVIDIA RTX 5090 GPU, on which the complete pre-training and fine-tuning pipeline finishes within one hour, comparable with the DDPM-based solvers (M4 and M5) and more than an order of magnitude faster than the PINN-based solvers (M2, $> 24\,\text{h}$). Although the supervised end-to-end solver (M1) can be trained within five minutes, it attains the worst accuracy and feasibility in Table 1.

Table 3: Comparison of Offline Training Time.

M1	M2	M3	M4	M5	PIDSolver
≤ 5 min	> 24 h	-	≤ 1 h	≤ 1 h	≤ 1 h

4.4 More Details Experiments

We conclude this section with a series of supplementary experiments that examine the ablation study, generalization, and statistical reliability of PIDSolver. Detailed results are reported in the Appendix D.

Sensitivity analysis. PIDSolver is robust over a wide range of the regularization coefficients: l_q attains its minimum at $\lambda_g = 0.1$ and degrades only when λ_g drops below 10^{-3}, a positive λ_c monotonically worsens feasibility without reducing the operating cost, and $\lambda_h = 0.1$ is optimal in the grid, so the default $(\lambda_h, \lambda_g, \lambda_c) = (0.1, 0.1, 0)$ is adopted. Detailed tables are reported in the Tables D.1 and D.2.

Reproducibility across training seeds. Training across five random seeds yields highly stable results, confirming that the reported gains do not rely on a lucky initialization (Table D.3).

Ablation on the physics-informed fine-tuning. The ablation study shows that the physics-informed fine-tuning of equation 10 reduces the reactive-power violation from 1.23×10^{-2} of the noise-only M4 baseline to 0.10×10^{-2} and eliminates the active power violations (Table D.4).

Out-of-distribution generalization. Under gradual load increments of up to $+23\%$ of the system load, PIDSolver generalizes gracefully out of distribution, maintaining 1.20–$4.5\times$ lower reactive violations than the frozen nominal dispatch while tracking the re-optimized OPF cost within 0.02–4.23% (Table D.5).

Statistical significance. Paired Wilcoxon signed-rank tests confirm that the improvements over the baselines are statistically significant, with $p < 10^{-160}$ (Table D.6).

5 Conclusion

This article introduces a physics-informed sequential diffusion solver (PIDSolver) for optimization problems with complex algebraic constraints. Inspired by traditional optimization methods, PIDSolver decomposes the solution process into multiple denoising steps that iteratively refine the candidate solution toward the optimum. Its core is a sequential architecture that couples a noise prediction network with an algebraic-constraint evaluation network: the latter replaces the iterative solver with an ANN surrogate, so that the gradients of the equality and inequality residuals propagate end-to-end and both training and inference become efficient. A physics-informed denoising loss jointly accounts for the optimization objective and the constraint violations, and an accelerated sampling strategy reduces the inference time to the order of seconds. From an energy-based perspective, we establish the fidelity of the ANN surrogate and a probabilistic feasibility guarantee of the generated solutions, and the residual energy of the reconstructed solutions decreases along the sampling path. Validation on the 118-bus test system shows that PIDSolver attains the lowest constraint violations among the learning-based methods, with an operating cost comparable to that of the reference solver and a modest and acceptable increase in computational cost. Although this paper focuses on AC-OPF, the proposed framework is applicable to other optimization problems with differential equality constraints. In future work, we will extend PIDSolver to large power systems.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks that require disclosure: formulating mathematical claims, assisting in the writing of mathematical proofs, providing feedback on research methodology and experimental design, assisting with translation. We have not used generative AI tools to develop theoretical models and conceptual frameworks, provide critical ingredients for claims, propose and refine hypotheses, generate synthetic data sets and interpret results, and the remaining required disclosure tasks (e.g., cleaning and reformatting datasets, supporting qualitative and thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools for the following tasks for which disclosure is recommended: editing the manuscript to improve readability, suggesting the structure of the paper and of its appendix. We have reviewed all AI-assisted work: mathematical claims and proofs were verified line by line and cross-checked against the experimental evidence, and AI-drafted text was revised by the authors for scientific accuracy and consistency. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ahmed Babiker, Sulaiman S Ahmad, Ijaz Ahmed, Muhammad Khalid, Mohammad A Abido, and Fahad Saleh Al-Ismail. Optimal power flow: A review of state-of-the-art techniques and future perspectives. *IEEE Access*, 13:60012–60039, 2025.

Kyri Baker. Emulating ac opf solvers with neural networks. *IEEE Transactions on Power Systems*, 37(6):4950–4953, 2022. doi: 10.1109/TPWRS.2022.3195097.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 3360–3385, 2025.

Kejun Chen, Shourya Bose, and Yu Zhang. Physics-informed gradient estimation for accelerating deep learning-based ac-opf. *IEEE Transactions on Industrial Informatics*, 21(6):4649–4660, 2025. doi: 10.1109/TII.2025.3545080.

Carleton Coffrin, Russell Bent, Kaarthik Sundar, Yeesian Ng, and Miles Lubin. Powermodels. jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8, 2018. doi: 10.23919/PSCC.2018.8442948.

V.M. da Costa, N. Martins, and J.L.R. Pereira. Developments in the newton raphson power flow formulation based on current injections. *IEEE Transactions on Power Systems*, 14(4):1320–1326, 1999. doi: 10.1109/59.801891.

Priya Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

Md Shamim Hasan, Md Mahmud-Ul-Tarik Chowdhury, and Sukumar Kamalasadan. Sequential quadratic programming (sqp) based optimal power flow methodologies for electric distribution system with high penetration of ders. *IEEE Transactions on Industry Applications*, 60(3):4810–4820, 2024. doi: 10.1109/TIA.2024.3371428.

Md Shamim Hasan, Md Mahmud-Ul-Tarik Chowdhury, and Sukumar Kamalasadan. Optimal power flow in electric transmission system: Comparative study and analysis of modeling methods. *IEEE Transactions on Industry Applications*, 61(3):4328–4341, 2025. doi: 10.1109/TIA.2025.3540819.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Milad Hoseinpour and Vladimir Dvorkin. Diffopf: Diffusion solver for optimal power flow, 2026. URL https://arxiv.org/abs/2510.14075.

Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H. Low. Deepopf-v: Solving ac-opf problems efficiently. *IEEE Transactions on Power Systems*, 37(1):800–803, 2022. doi: 10.1109/TPWRS.2021.3114092.

Wanjun Huang, Minghua Chen, and Steven H. Low. Unsupervised learning for solving ac optimal power flows: Design, analysis, and experiment. *IEEE Transactions on Power Systems*, 39(6): 7102–7114, 2024. doi: 10.1109/TPWRS.2024.3373399.

IEA. Electricity market design. Technical report, International Energy Agency, 2025. URL https://www.iea.org/reports/electricity-market-design. Accessed: 2026-09-21.

Sandro Merkli, Alexander Domahidi, Juan L. Jerez, Manfred Morari, and Roy S. Smith. Fast ac power flow optimization using difference of convex functions programming. *IEEE Transactions on Power Systems*, 33(1):363–372, 2018. doi: 10.1109/TPWRS.2017.2688329.

Sina Mohammadi, Van-Hai Bui, Wencong Su, and Bin Wang. Surrogate modeling for solving opf: A review. *Sustainability*, 16(22):9851, 2024.

Xiang Pan, Tianyu Zhao, and Minghua 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)*, pp. 1–6, 2019. doi: 10.1109/SmartGridComm.2019.8909795.

Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H. Low. Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems. *IEEE Systems Journal*, 17(1): 673–683, 2023. doi: 10.1109/JSYST.2022.3201041.

Shashank Shekhar, Abhinav Karn, Kris Keshav, Shivam Bansal, and Parikshit Pareek. Fast diffusion with physics-correction for acopf. *arXiv preprint arXiv:2602.03020*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1): 25–57, 2006.

A Proofs

A.1 Proof of Lemma 1

Since $\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}^\dagger) = \mathbf{0}$, the L_h-Lipschitz continuity of $\mathbf{h}$ in $\mathbf{z}$ yields $\left\|\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \hat{\mathbf{z}})\right\|_2 = \left\|\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \hat{\mathbf{z}}) - \mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}^\dagger)\right\|_2 \le L_h \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\|_2 \le L_h \epsilon_z$. Similarly, the L_g-Lipschitz continuity of $\mathbf{g}$ gives $\left\|\mathbf{g}(\mathbf{x}_0, \hat{\mathbf{z}}) - \mathbf{g}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2 \le L_g \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\|_2 \le L_g \epsilon_z$. Combining the two bounds, $\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}}) - \boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2^2 = \left\|\mathbf{h}(\mathbf{x}_0, \mathbf{y}, \hat{\mathbf{z}}) - \mathbf{h}(\mathbf{x}_0, \mathbf{y}, \mathbf{z}^\dagger)\right\|_2^2 + \left\|\mathbf{g}(\mathbf{x}_0, \hat{\mathbf{z}}) - \mathbf{g}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2^2 \le (L_h^2 + L_g^2)\epsilon_z^2$, which proves the first two claims. For the last claim, the mean squared error term $\mathbb{E}\left[\|\hat{\mathbf{z}} - \mathbf{z}\|_2^2\right]$ of the pre-training objective $\mathcal{L}(\boldsymbol{\Theta}_2)$ drives the expected surrogate error ϵ_z toward zero, while the power flow imbalance term $\eta\|\boldsymbol{\mathcal{R}}_{\mathbf{h}}\|_2^2$ further regularizes the surrogate toward points satisfying $\mathbf{h} = \mathbf{0}$.

A.2 Proof of Theorem 1

By Lemma 1, $\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}}) - \boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2 \le \sqrt{L_h^2 + L_g^2}\,\epsilon_z$, so the triangle inequality gives $\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2 \le \left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}})\right\|_2 + \sqrt{L_h^2 + L_g^2}\,\epsilon_z$. Taking expectations and applying Jensen's inequality to the premise yields $\mathbb{E}_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left[\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2\right] \le \sqrt{\mathbb{E}\left[\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \hat{\mathbf{z}})\|_2^2\right]} + \sqrt{L_h^2 + L_g^2}\,\epsilon_z \le \delta + \sqrt{L_h^2 + L_g^2}\,\epsilon_z$. Markov's inequality then gives $P_{\mathbf{x}_0 \sim p_{\boldsymbol{\Theta}}}\left(\left\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\right\|_2 > \varepsilon\right) \le \frac{\mathbb{E}\left[\|\boldsymbol{\mathcal{R}}(\mathbf{x}_0, \mathbf{z}^\dagger)\|_2\right]}{\varepsilon} \le \frac{\delta + \sqrt{L_h^2 + L_g^2}\,\epsilon_z}{\varepsilon}$, which proves the theorem.

B TRAJECTORY VISUALIZATIONS OF THE ILLUSTRATIVE EXAMPLES

This appendix collects the denoising trajectories of the three illustrative examples of Section 4. The three cases increase in difficulty—from an unconstrained problem, to box constraints, to an implicit algebraic constraint solved through an ANN surrogate—thereby mirroring the ingredients of the sequential PIDSolver architecture at a miniature scale. Each figure follows the same six-panel layout: (a) forward noising trajectories starting from the optimal solution; (b) reverse denoising trajectories converging to the target; (c) noisy (dashed) versus denoised (solid) trajectories on the objective surface; (d) XY-plane projection with the infeasible region shaded; (e) objective-value curves in log scale; and (f) distance-to-target curves in log scale.

Case 1: Unconstrained baseline. A plain DDPM is trained in the state space $(x, y, f(x, y))$, where the third coordinate is the objective value $f(x, y)$ and serves only for visualization, to generate the unconstrained optimum $(x^*, y^*) = (2, 3)$ with $f^* = 0$. Panels (b) and (f) show that the reverse process reconstructs this optimum from arbitrary initializations, verifying the generative capability of the plain diffusion model for unconstrained optimization.

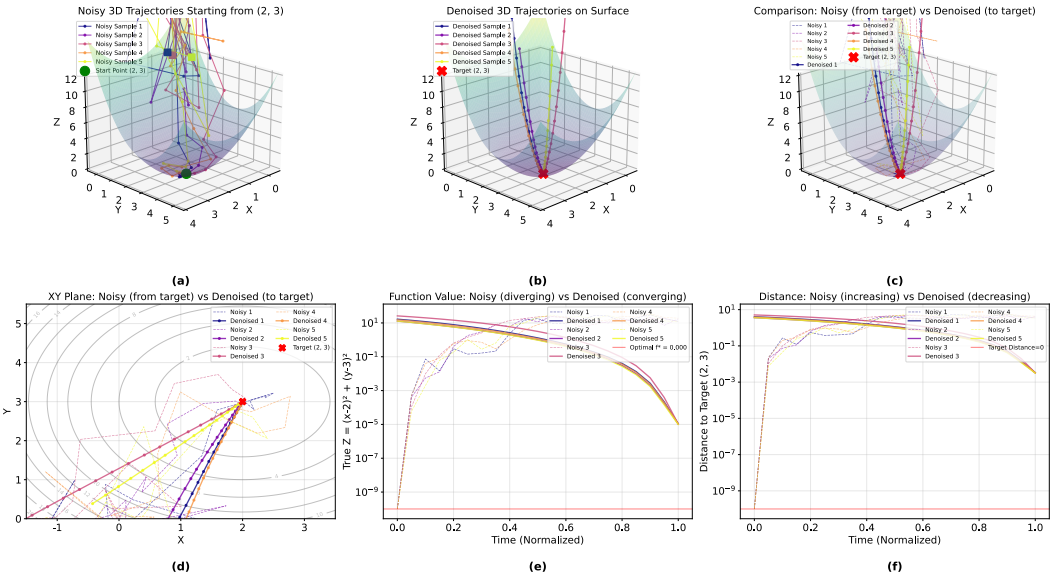

Figure B.1: Case 1 (unconstrained baseline): the optimum is $(2, 3)$ with $f^* = 0$.

Case 2: Inequality constraints. Adding the box constraints $x \leq 1$ and $y \leq 4$ shifts the optimum to $(x^*, y^*) = (1, 3)$ with $f^* = 1$, where, as in the unconstrained baseline, the third diffusion coordinate is the objective value $f(x, y)$. Following the physics-informed objective in equation 10, the DDPM is trained with a constraint-violation penalty evaluated on the reconstructed clean solution: $\mathcal{L} = \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}\right\|_2^2 + \lambda \left[\mathrm{relu}\big(\hat{x}_0 - 1\big)^2 + \mathrm{relu}\big(\hat{y}_0 - 4\big)^2\right]$. The penalty acts as an energy barrier: in panels (c) and (d) the reverse trajectories are visibly steered into the shaded feasible region, converging to $(1, 3)$ while excluding the unconstrained attractor $(2, 3)$.

Case 3: Implicit algebraic constraint with an ANN surrogate. The third case couples the decision variables through the implicit algebraic constraint $x = \sin(z) + \cos(y)$ together with $x \leq 1$, $y \leq 4$, $z \leq 2$; the optimum is $(x^*, y^*, z^*) = (1, \pi/2, \pi/2)$ with $f^* = 1 + (3 - \pi/2)^2 \approx 3.04$. Here the third diffusion coordinate is the coupled variable z itself, which plays the role of the latent variables $\mathbf{z}$ solved by the AC-PF evaluation network in the main framework: the DDPM generates the candidate (x, y, z), while a small ANN surrogate predicts $z^* = z_{\boldsymbol{\Theta}_2}(x, y)$ and evaluates the equality residual during denoising. The PIDSolver objective augments the denoising term with explicit inequality penalties and the equality residual evaluated through the surrogate: $\mathcal{L}_{\text{PIDSolver}} = \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\Theta}_1}\right\|_2^2 + \lambda_g \left[\mathrm{relu}(x - 1)^2 + \mathrm{relu}(y - 4)^2 + \mathrm{relu}\big(z^* - 2\big)^2\right] + \lambda_h \left[\big(\sin(z^*) + \cos(y) - x\big)^2\right]$. Because the equality residual is evaluated through the surrogate, the trajectories in panels (c) and (d) are guided along the implicit constraint surface toward the feasible optimum $(1, \pi/2, \pi/2)$.

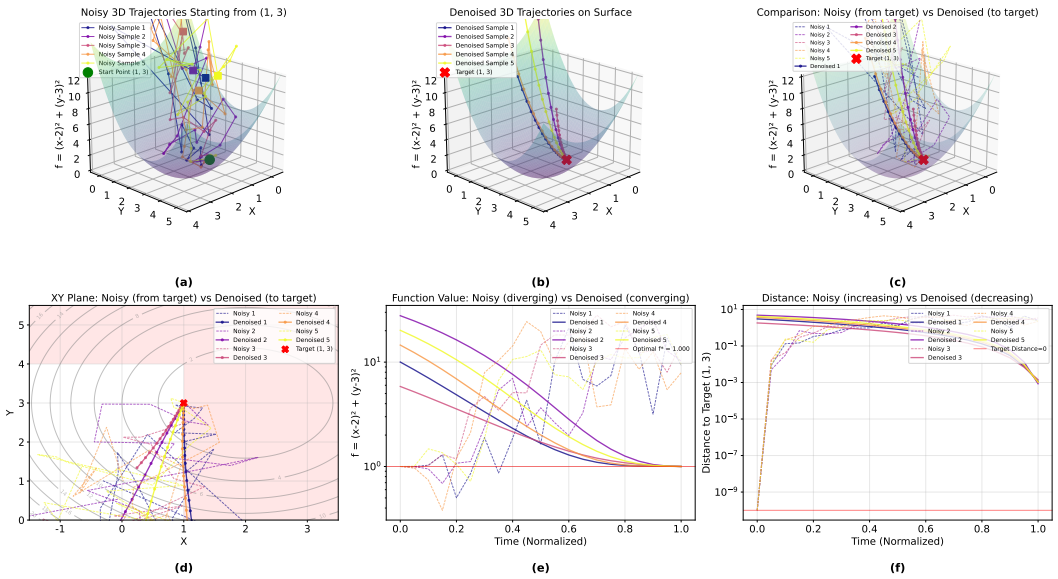

Figure B.2: Case 2 (inequality constraints, $x \leq 1$, $y \leq 4$): the constrained optimum is $(1, 3)$ with $f^* = 1$.

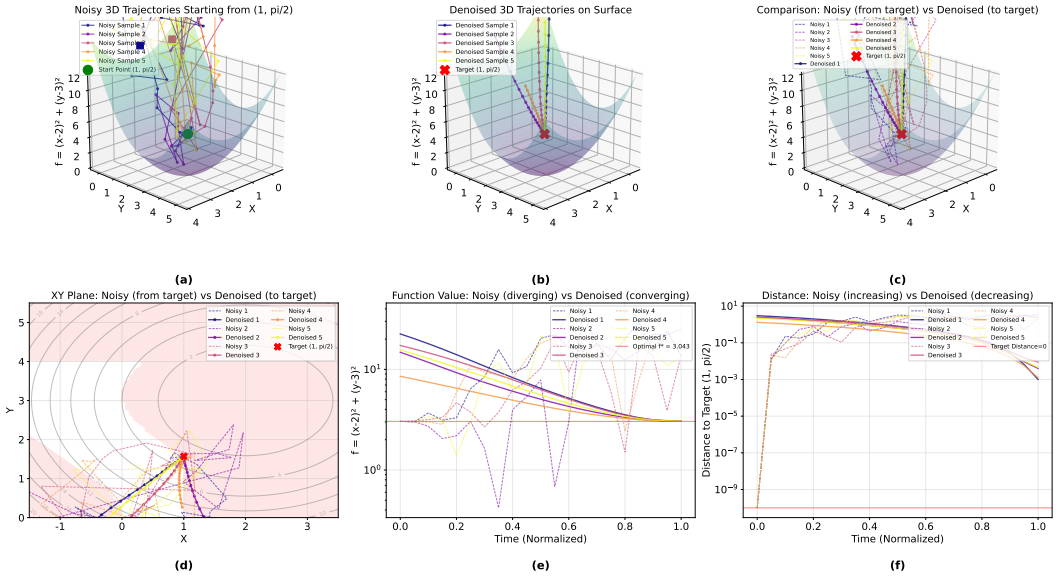

Figure B.3: Case 3 (implicit algebraic constraint, $x = \sin(z) + \cos(y)$, $x \leq 1$, $y \leq 4$, $z \leq 2$): the feasible optimum is $(x^*, y^*, z^*) = (1, \pi/2, \pi/2)$ with $f^* = 1 + (3 - \pi/2)^2 \approx 3.04$.

C Model Structure, Training Parameters and Capacity Sensitivity

Table C.1 reports the sensitivity to the size of the DDPM backbone, parameterized by its hidden width h_d and depth n_l (number of residual blocks). The default backbone $(512, 6)$ attains the lowest reactive violation ($l_q = 0.10 \times 10^{-2}$ over five seeds); enlarging it to $(1024, 8)$ brings no obvious gain. Reducing the depth alone, $(512, 3)$, degrades l_q to 0.14×10^{-2}, and reducing the width to $(256, 4)$ degrades it to 0.20×10^{-2}, indicating that residual depth contributes more to the generative quality than width. All backbones maintain 100% AC-PF convergence and a cost gap below 0.005%, so the default configuration sits at the best capacity-accuracy trade-off.

Table C.1: Sensitivity to the Size of the DDPM Backbone (Five-Seed Mean).

(h_d, n_l)	l_{cost} (%)	Constraint Violations (10^{-2})				PF Conv.
		$\bar{l}_v$	l_p	l_q	l_u	
(256, 4)	0.01	0.03	0.00	0.20	0.00	100%
(512, 6)	0.01	0.01	0.00	**0.10**	0.00	100%
(1024, 8)	0.01	0.01	0.00	0.09	0.00	100%
(512, 3)	0.01	0.02	0.00	0.14	0.00	100%

Table C.2 reports the sensitivity to the internal width d_{int} of the AC-PF evaluation network, in which the pre-trained DDPM is shared across variants and only the pre-training and fine-tuning of the surrogate are retrained. Under the regularized surrogate pre-training objective $\mathcal{L}(\mathbf{\Theta}_2)$, the width has negligible effect: l_q stays within 0.08–0.09×10^{-2} for $d_{int} \in \{16, 32, 64\}$. Together, the two sweeps justify the adopted default configuration—a $(512, 6)$ DDPM backbone with $d_{int} = 32$—as a robust operating point.

Table C.2: Sensitivity to the Width d_{int} of the AC-PF Evaluation Network.

d_{int}	l_{cost} (%)	Constraint Violations (10^{-2})				PF Conv.
		$\bar{l}_v$	l_p	l_q	l_u	
16	0.01	0.01	0.00	0.08	0.00	100%
32 (default)	0.01	0.01	0.00	**0.09**	0.00	100%
64	0.01	0.01	0.00	0.08	0.00	100%

The final model architecture and the training hyper-parameters of the adopted configuration are summarized in Table C.3, where LN refers to the layernorm and d refers to the input shape of residual blocks.

Table C.3: Structure and training parameters of the proposed PIDSolver

Attributes	Layers	Parameters	Activation
Noise Prediction Network	Action Branch	units: [108, 512, 5 Residual Blocks, 512]	SiLU
	State Branch	units: [236, 512, 5 Residual Blocks, 512]	SiLU
	Time embedding	units: [1, 128, 128, 128]	SiLU
	Merge	units: [1152, 512, 5 Residual Blocks, 512, 108]	Linear
	Residual Blocks	units: [d, $2d$, LN, $2d$, d]	-
AC-PF Evaluation Network	Action Branch	units: [108, 32, 64, 128, 256, 512, 1024, 256]	SiLU
	State Branch	units: [236, 32, 64, 128, 256, 512, 1024, 256]	SiLU
	Merge	units: [512, 512, 1024, 2048, 1024, 236]	SiLU & Sigmoid
Hyper-parameters	Batch Size	Epochs	Learning Rate Scheduler
Pre-training, noise prediction network	512	4000	Cosine Scheduler
Pre-training, AC-PF evaluation network	512	40	Cosine Scheduler
Fine-tuning, AC-PF evaluation network	512	50	Cosine Scheduler
Fine-tuning, sequential network (joint)	512	≤50 epochs, 2 batches per epoch, early stopping on validation set	Linear Scheduler

Notes: the pre-training loss of the AC-PF evaluation network is MSE $+ \eta\|\boldsymbol{\mathcal{R}}_{\mathbf{h}}\|^2$ ($\eta = 0.01$); the joint fine-tuning loss is equation 10 with $(\lambda_h, \lambda_g, \lambda_c) = (0.1, 0.1, 0)$, $S = 5$ refinement steps and the full end-to-end gradient of equation 11–equation 12.

D Details Experiments

D.1 Sensitive Analysis

We investigate how the regularization coefficients affect the performance of PIDSolver. Table D.1 sweeps the inequality weight λ_g over $\{1, 0.1, 0.01, 0.001, 0.0001\}$ with three training seeds per value ($\lambda_h = 0.1$, $\lambda_c = 0$): the reactive violation l_q attains its minimum at $\lambda_g = 0.1$ ($0.10 \pm 0.01 \times 10^{-2}$), degrades modestly for larger or smaller values. Table D.2 sweeps $\lambda_h \in \{0, 0.01, 0.1\}$

against $\lambda_c \in \{0, 0.001, 0.01\}$: a positive λ_c monotonically degrades feasibility without lowering the operating cost (the cost gap is already below 0.005%), and $\lambda_h = 0.1$ attains the lowest l_q in the grid. The final configuration therefore uses $(\lambda_h, \lambda_g, \lambda_c) = (0.1, 0.1, 0)$.

Table D.1: Sensitive Analysis (T = 1000).

λ_g	l_{cost} (%)	Constraint Violations (10^{-2})			
		$\bar{l}_v$	l_p	l_q	l_u
1	0.01±0.00	0.02±0.01	0.00±0.00	0.14±0.06	0.00±0.00
0.1	0.01±0.00	0.01±0.00	0.00±0.00	**0.10**±0.01	0.00±0.00
0.01	0.00±0.00	0.01±0.00	0.00±0.00	0.10±0.01	0.00±0.00
0.001	0.00±0.00	0.03±0.00	0.00±0.00	0.21±0.02	0.00±0.00
0.0001	0.00±0.00	0.03±0.00	0.00±0.00	0.23±0.01	0.00±0.00

Table D.2: Grid over the equality and cost weights (λ_h, λ_c) at $\lambda_g = 0.1$.

$\lambda_h \setminus \lambda_c$	0	0.001	0.01
0	0.019	0.020	0.025
0.01	0.023	0.024	0.046
0.1	**0.015**	0.016	0.028

D.2 Reproducibility Across Training Seeds

Table D.3 reports the statistics of the five training seeds. The reactive violation l_q varies within the narrow interval $[0.09, 0.12] \times 10^{-2}$ (mean 0.10 ± 0.01), and the remaining metrics exhibit even smaller dispersion.

Table D.3: Reproducibility Across Five Training Seeds (PIDSolver, $T = 1000$).

Statistic	l_{cost} (%)	$\bar{l}_v$	l_p	l_q	l_u
Mean	0.01	0.01	0.00	0.10	0.00
Std.	<0.001	0.00	0.00	0.01	0.00
Min	0.01	0.01	0.00	0.09	0.00
Max	0.01	0.02	0.00	0.12	0.00

D.3 Ablation on the Physics-informed Fine-tuning

Table D.4 evaluates the facts that determine the effectiveness of the physics coupling. Training with the noise-prediction loss alone (M4) yields $l_q = 1.14 \times 10^{-2}$. The physics-informed fine-tuning of equation 10 reduces l_q to 0.10×10^{-2} over five seeds and eliminates the active power violations. With the balanced loss equation 10—in which the equality and inequality residuals are weighted separately and the surrogate is pre-trained with the regularized objective $\mathcal{L}(\mathbf{\Theta}_2)$ of the pre-training stage—the *full* gradient derived in equation 11 and equation 12, is stable and attains the lowest violations.

D.4 Out-of-Distribution Generalization

Table D.5 evaluates the methods under gradual load increments, where δ MW is added to the active load of each of the 99 load buses (a total of 99δ MW, i.e., up to $+23\%$ of the system load). For every perturbed load profile, the AC-OPF is re-solved with PowerModels.jl (the "OPF (re-solved)" row), which yields the true optimal cost $C^*(\delta)$ and is feasible by construction. At the nominal operating point ($\delta = 0$), all methods are near-optimal. As the load shifts, the frozen nominal dispatch (PowerModel (nominal)) degrades steeply. Because PIDSolver conditions on the perturbed load, its violations grow far more gracefully, and its operating cost tracks the re-optimized optimum within

Table D.4: Ablation on the Physics-informed Fine-tuning (λ_h=0.1, λ_g=0.1, λ_c=0; $T = 1000$).

Configuration	l_{cost} (%)	Constraint Violations (10^{-2})			
		$\bar{l}_v$	l_p	l_q	l_u
M4	0.01	0.16	0.02	1.14	0.00
Full gradient (default), 2 batches	**0.01**	**0.01**	**0.00**	**0.10**	**0.00**
Full gradient, all batches	0.01	0.01	0.00	0.06	0.00

$+0.02\%$, $+0.05\%$, $+0.57\%$, and $+4.23\%$ at $\delta = 1, 2, 5, 10$ MW. The physics fine-tuning is the most important ingredient: compare to M4, it reduces l_q by $10.8\times$ at $\delta = 1$ MW and by $4.4\times$ at $\delta = 5$ MW, and it keeps the active violation $3.2\times$ below the nominal dispatch at $\delta = 10$ MW.

Table D.5: Out-of-Distribution Generalization under Gradual Load Increments.

δ	Methods	l_{cost} (%)	Constraint Violations (10^{-2})			
			$\bar{l}_v$	l_p	l_q	l_u
0 MW	PIDSolver	0.01	0.01	0.00	**0.10**	0.00
	M4	0.01	0.15	0.02	1.14	0.00
	PowerModel (nominal)	0.00	0.00	0.00	0.00	0.00
	OPF (re-solved)	0.00	0.00	0.00	0.00	0.00
1 MW	PIDSolver	0.02	0.01	0.00	**0.10**	0.00
	M4	0.02	0.15	0.01	1.12	0.00
	PowerModel (nominal)	0.17	0.03	0.00	0.21	0.00
	OPF (re-solved)	0.00	0.00	0.00	0.00	0.00
2 MW	PIDSolver	0.05	0.01	0.00	**0.11**	0.00
	M4	0.07	0.15	0.01	1.12	0.00
	PowerModel (nominal)	0.69	0.06	0.00	0.47	0.00
	OPF (re-solved)	0.00	0.00	0.00	0.00	0.00
5 MW	PIDSolver	0.57	0.05	0.00	**0.35**	0.00
	M4	0.58	0.20	0.01	1.53	0.00
	PowerModel (nominal)	4.12	0.66	3.50	1.52	0.00
	OPF (re-solved)	0.00	0.00	0.00	0.00	0.00
10 MW	PIDSolver	4.23	1.03	**4.48**	**3.36**	0.00
	M4	4.21	1.23	4.45	4.93	0.00
	PowerModel (nominal)	15.57	2.39	14.21	4.02	0.00
	OPF (re-solved)	0.00	0.00	0.00	0.00	0.00

D.5 Statistical Significance

Table D.6 reports paired Wilcoxon signed-rank tests on the per-sample reactive violations. PIDSolver is significantly better than M4 ($p < 10^{-160}$), confirming that the reported gaps are not artifacts of the sampled test set.

Table D.6: Paired Wilcoxon Signed-Rank Tests ($n = 1000$ samples, per-sample l_q).

PIDSolver vs.	Statistic	p-value
M4	0.0	3.3×10^{-165}