NEO-Grid: A Neural Approximation Framework for Optimization and Control in Distribution Grids

Mohamad Chehade and Hao Zhu
Chandra Family Department of Electrical and Computer Engineering
The University of Texas at Austin
{chehade, haozhu}@utexas.edu

Abstract

The rise of distributed energy resources (DERs) is reshaping modern distribution grids, introducing new challenges in attaining voltage stability under dynamic and decentralized operating conditions. This paper presents NEO-Grid, a unified learning-based framework for volt-var optimization (VVO) and volt-var control (VVC) that leverages neural network surrogates for power flow and deep equilibrium models (DEQs) for closed-loop control. Our method replaces traditional linear approximations with piecewise-linear ReLU networks trained to capture the nonlinear relationship between power injections and voltage magnitudes. For control, we model the recursive interaction between voltage and inverter response using DEQs, allowing direct fixed-point computation and efficient training via implicit differentiation. We evaluated NEO-Grid on the IEEE 33-bus system, demonstrating that it significantly improves voltage regulation performance compared to standard linear and heuristic baselines in both optimization and control settings. Our results establish NEO-Grid as a scalable, accurate, and interpretable solution for learning-based voltage regulation in distribution grids.

1. Introduction

The modern distribution grid is undergoing a rapid transformation driven by the proliferation of distributed energy resources (DERs) such as solar photovoltaics, electric vehicles, and battery storage. This evolution introduces new challenges for voltage regulation, as increased variability and decentralized injections can lead to voltage violations that compromise equipment safety and power quality. Among available solutions, *volt-var management*—the practice of modulating reactive power injections to regulate voltage magnitudes—has emerged as a key mechanism for maintaining voltage stability in the face of these disturbances [1].

There are two principal modes of implementing volt-var strategies. In **volt-var optimization (VVO)**, reactive power injections are computed via centralized optimization to minimize system-wide voltage deviation [2]. In contrast, **volt-var control (VVC)** seeks to learn local, autonomous control policies that map bus-level voltage measurements to reactive power decisions [3]. Both approaches require accurate modeling of how voltages respond to power injections—i.e., the power flow (PF) relationship.

However, PF models are governed by nonlinear and non-convex equations derived from Kirchhoff's laws. These models are difficult to embed directly in optimization pipelines due to their computational cost and numerical instability. To address this, many recent works adopt linear approximations, such as the LinDistFlow model [4; 5], which express voltages as affine functions of power injections. While these approximations enable tractable optimization, they often fail to capture key nonlinearities, especially under heavy loading or highly resistive lines.

To bridge this gap, we adopt a data-driven modeling approach using piecewise-linear neural networks. These ReLU-based models serve as flexible surrogates that can approximate the complex, nonlinear power flow mappings with high accuracy [6; 7]. Crucially, they are differentiable and easily integrated into optimization frameworks, enabling neural approximations to serve as drop-in replacements for analytical models in tasks like VVO. For instance, by formulating the learned network as a set of mixed-integer linear constraints, the resulting optimization can faithfully replicate the physical behavior of the system without requiring access to full network parameters—a principle also explored in recent constraint-learning approaches [8].

Nonetheless, while VVO benefits directly from neural PF surrogates, the design of optimal VVC rules presents a deeper challenge. The interaction between voltage and control introduces a recursive dynamic: each inverter measures local voltage, applies a control rule, which then alters the voltage state of the network, which in turn affects subsequent decisions. This closed-loop behavior forms a dynamical system that must be simulated or optimized over—posing signifi-

cant challenges for gradient-based learning. Prior work has attempted to address this by unrolling the recursion through recurrent neural networks (RNNs) [9], but such approaches suffer from vanishing gradients and memory bottlenecks.

To address this core limitation, we adopt the framework of **deep equilibrium models (DEQs)** [10; 11]. Rather than unrolling the recursive system over multiple time steps, DEQs solve directly for the fixed-point voltage profile induced by the control rule. This enables stable, memory-efficient learning of closed-loop policies while preserving the physical dynamics of the system. Furthermore, DEQs support implicit differentiation: gradients are computed through the equilibrium point itself without backpropagating through the full trajectory. This allows us to train optimal control rules end-to-end in a scalable and principled manner.

Our proposed framework, termed `NEO-Grid` (NEural-based Optimization and control of Distribution Grids), unifies data-driven modeling and control for distribution voltage regulation. Unlike prior methods, such as [3; 9], which rely on linear approximations and RNN-based control, NEO-Grid is both nonlinear and equilibrium-based—delivering enhanced modeling fidelity and training stability.

Contributions. The main contributions of this work are:

1. We introduce a unified framework for learning-based distribution grid optimization, encompassing both volt-var optimization (VVO) and optimal volt-var control (VVC).

2. We develop neural surrogate models for nonlinear power flow and use them for both optimization and control, improving accuracy over traditional linear or heuristic methods.

3. We apply deep equilibrium models (DEQs) to power system control for the first time, enabling efficient learning of optimal, closed-loop control rules.

2. System Modeling

We consider a distribution system consisting of a substation bus feeding N buses in the set $\mathcal{N} = \{1, \ldots, N\}$. We assume a radial, single-phase system for simplicity, and thus the number of lines is also N. However, the proposed machine learning and optimized decision making techniques can be generalized to non-radial, multi-phase networks, as well. A subset of buses in $\mathcal{D} \subset \mathcal{N}$ hosts distributed energy resources (DERs), such as inverter-interfaced generation or battery. Each node $i \in \mathcal{N}$ is connected to the active and reactive power demand indicated by (p_i^c, q_i^c) and generation (p_i^g, q_i^g). In addition, let v_i denote the voltage magnitude per bus i, and assume that the substation voltage v_0 is fixed and known. We can stack all voltage/power variables into $N \times 1$ vectors $\mathbf{v}$, $\mathbf{p}$, and $\mathbf{q}$, with the injection $\mathbf{p} := \mathbf{p}^g - \mathbf{p}^c$, and similarly for $\mathbf{q}$.

Let us use $\mathbf{v} = f(\mathbf{p}, \mathbf{q})$ to represent the nonlinear AC-PF model. To form $f(\cdot)$, we need to know the the network topology given by the set of N line segments in $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$, as well as the resistance r_{ij} and reactance x_{ij} for each line $(i, j) \in \mathcal{E}$. Let p_{ij} and q_{ij} respectively denote the real and reactive power flowing from bus i to bus j on line (i, j). This allows one to express the nonlinear PF model $f(\cdot)$ using the well-known DistFlow equations [12]:

$$p_{ij} = \sum_{k \in \mathcal{N}_j} p_{jk} + r_{ij} \frac{p_{ij}^2 + q_{ij}^2}{v_i^2} + p_j^g - p_j^c, \tag{1a}$$

$$q_{ij} = \sum_{k \in \mathcal{N}_j} q_{jk} + x_{ij} \frac{p_{ij}^2 + q_{ij}^2}{v_i^2} + q_j^g - q_j^c, \tag{1b}$$

$$v_j^2 = v_i^2 - 2(r_{ij} p_{ij} + x_{ij} q_{ij}) + (r_{ij}^2 + x_{ij}^2) \frac{p_{ij}^2 + q_{ij}^2}{v_i^2}. \tag{1c}$$

Here, $\mathcal{N}_j$ denotes the set of downstream child nodes of bus j. The fractional term $(p_{ij}^2 + q_{ij}^2)/v_i^2$ represents the squared line current magnitude and is associated with power losses. To simplify the analysis, the linearized Distflow (LinDistFlow) model is widely used, which ignores line power losses. This will be discussed more in Section 3 later on.

Optimizing feeder voltage using the $\mathbf{q}^g$ flexibility from smart inverters along with other reactive power (var) resources is becoming increasingly important to mitigate the volatility of distribution systems. We consider a basic version of the **volt-var optimization (VVO)** problem [2; 3; 5], that seeks to minimize the voltage deviation from the nominal 1.0 pu given by

$$\min_{\mathbf{q}^g} \quad \|\mathbf{v} - \mathbf{1}\|_2^2 \tag{2a}$$

$$\text{s.t.} \quad \mathbf{v} = f(\mathbf{p}, \mathbf{q}^g - \mathbf{q}^c) \tag{2b}$$

$$\underline{\mathbf{q}} \le \mathbf{q}^g \le \bar{\mathbf{q}} \tag{2c}$$

Here, the range $[\underline{\mathbf{q}}, \bar{\mathbf{q}}]$ relates to the var limits that can be supplied or absorbed by inverters, as dictated by their physical limits or operational standards. For simplicity, the rest of paper uses $\mathbf{q}$ to denote the controllable portion of reactive power (i.e., $\mathbf{q}^g$), not the total injection. The VVO formulation can be extended to include power losses in the objective function or to enforce volt-

age regulation constraints to precisely satisfy the standard limits of $\pm 5\%$ voltage deviations. It can also be generalized to include the possible flexibility in $\mathbf{p}$, or encompass various optimization tasks for distribution management system; see e.g., a recent overview paper [1].

Moreover, the VVO problem (2) is the basis for designing the optimal **volt-var control (VVC)** rules [9; 13]. Different from VVO that searches any feasible $\mathbf{q}$, the VVC problem seeks a decision rule of $q_i \leftarrow g(v_i; \mathbf{z}_i)$ that is parameterized by $\mathbf{z}_i$, such that the q_i decision at bus i can be continuously updated based only on the localized voltage v_i. This localized design greatly reduces computational and communication overhead, and thus is easy to implement in practice. We will discuss more details on the design of the VVC decision rules in Section 4.

For solving either of these two problems, the key challenge lies in the nonlinearity of the AC-PF model $f(\cdot)$. The next section will present our neural approximation approach to address this challenge by developing accurate yet computationally tractable PF models.

3. `NEO-Grid`: neural PF approximation

The gist of our proposed `NEO-Grid` is to develop an accurate yet tractable approximation for AC power flow (AC-PF) models. The following provides a comprehensive overview of both linear and neural approximation approaches relevant to distribution-level optimization and control tasks.

Linear PF Approximation: To simplify the AC-PF model, linearized approximations are widely used. In particular, the LinDistFlow model approximates the nonlinear equations (1) by neglecting loss terms and linearizing around $v_i \approx 1.0$ pu using $v_i^2 \approx 2v_i - 1$, yielding:

$$\mathbf{v} = \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + \mathbf{1}v_0$$

Here, $\mathbf{R}$ and $\mathbf{X}$ are sensitivity matrices derived from feeder topology and line impedance parameters. For the VVO problem in (2), the model is often written as $\mathbf{v} = \mathbf{X}\mathbf{q}^g + \bar{\mathbf{v}}$, where $\bar{\mathbf{v}}$ denotes the nominal voltage profile when no reactive flexibility is available.

To improve accuracy, data-driven linearization can be employed. Given training samples $\{[\mathbf{p}; \mathbf{q}], \mathbf{v}\}$, a least-squares (LS) regression model is trained by minimizing the loss:

$$\|\mathbf{v} - f^{\mathrm{LS}}([\mathbf{p}; \mathbf{q}]; \boldsymbol{\beta})\|_2^2$$

where $\boldsymbol{\beta}$ are the linear (affine) model coefficients. The target output can also be chosen as voltage deviations from nominal, i.e., $(\mathbf{v} - \bar{\mathbf{v}})$. While linear approximations greatly simplify downstream optimization and control, their accuracy diminishes under substantial voltage drops, high losses, and diverse operating conditions.

NN-Based PF Approximation: Neural approximations improve on linear models by learning the nonlinear mapping between power injections and voltages from data. Traditionally, piecewise-linear (PWL) approximations have been proposed to capture nonlinearities by partitioning the input space into multiple linear regions [14]. However, PWL methods require manual region partitioning or mixed-integer formulations, which are often heuristic or computationally intensive.

Neural networks (NNs) offer a scalable alternative: as universal function approximators, they can flexibly learn the nonlinear structure directly from data without explicit partitioning. The trainable ReLU activations in the hidden layers effectively encode the system's piecewise-linear and nonlinear characteristics.

To train the NN approximation $f^{\mathrm{NN}}(\cdot)$, we generate datasets by perturbing nominal consumption profiles to create diverse samples of net active and reactive power $[\mathbf{p};\ \mathbf{q}]$ along with corresponding voltages $\mathbf{v}$ from power flow simulations. This ensures the dataset covers both typical and extreme operating scenarios, including DER compensation with negative reactive power.

Figure 1 illustrates the one-hidden-layer NN architecture employed in `NEO-Grid`. The input is the $2N$-dimensional net power injection vector, and the output is the N-dimensional voltage vector. The architecture consists of a hidden layer with K ReLU units and is parameterized by weight matrices $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and bias vector $\boldsymbol{\beta}$. Formally, the NN computes:

$$\mathbf{v} = \mathbf{W}^{(2)} \cdot \mathrm{ReLU}\big(\mathbf{W}^{(1)}[\mathbf{p}; \mathbf{q}] + \boldsymbol{\beta}\big)$$

where all parameters $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \boldsymbol{\beta}\}$ are learned from data.

Remark 1 (On Output Representations)**.** *While our model directly predicts the voltage vector* $\mathbf{v}$*, other forms of* $f^{\mathrm{NN}}(\cdot)$ *such as voltage deviations* $|\mathbf{v}-\mathbf{1}|$ *or the scalar objective* $\|\mathbf{v}-\mathbf{1}\|_2^2$ *have been considered in the literature for specific control objectives.*

Remark 2 (Compact NN Design for `NEO-Grid`)**.** *Although we focus here on a fully connected NN, it is advantageous to exploit the sparse feeder topology to build compact NN models with fewer parameters. Such topology-aware designs, as advocated in [6; 15], improve generalization and reduce overfitting—an avenue we will explore in future work.*

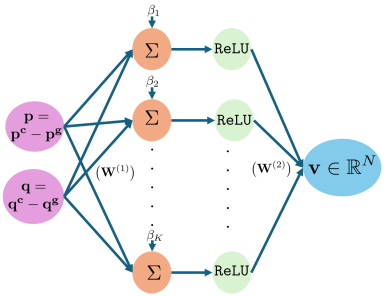

Figure 1: One-hidden-layer NN for approximating the mapping from net power injections $[\mathbf{p}; \mathbf{q}]$ to bus voltages $\mathbf{v}$. The trainable weights and biases are learned from data.

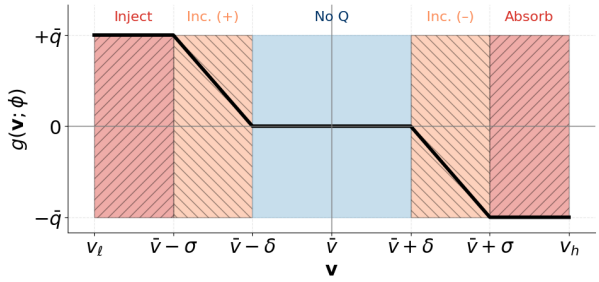

Figure 2: Illustration of the PWL volt-var control rule mapping $\mathbf{v}$ to $\mathbf{q}^g$. The operating regions include deadband, linear ramps, and saturation, parameterized by $\bar{v}$, δ, σ, and $\bar{q}$.

3.1 VVO on `NEO-Grid` via mixed-integer programming

Integrating a neural approximation $f^{\mathrm{NN}}(\cdot)$ into volt-var optimization (VVO) has emerged as a promising paradigm for scalable and data-driven voltage regulation; see e.g., [2; 8; 6]. The idea is to replace the nonlinear AC power flow (PF) equations with a learned piecewise-linear neural network (PWL-NN) surrogate model. This surrogate maps net active and reactive power injections directly to bus voltages, enabling the embedding of power flow constraints into an optimization problem without requiring explicit network parameters or solving non-convex equations.

The VVO problem is then formulated as:

$$\min_{\mathbf{q}^g} \quad \|\mathbf{v} - \mathbf{1}\|_2^2 \tag{3a}$$

$$\text{s.t.} \quad \mathbf{v} = f^{\mathrm{NN}}(\mathbf{p}^g - \mathbf{p}^c, \mathbf{q}^g - \mathbf{q}^c) \tag{3b}$$

$$\underline{\mathbf{q}}^g \le \mathbf{q}^g \le \overline{\mathbf{q}}^g \tag{3c}$$

where $\mathbf{v}$ is the bus voltage vector and $\underline{\mathbf{q}}^g, \overline{\mathbf{q}}^g$ denote the lower and upper limits on reactive power injection at the controllable nodes. The quantities $\mathbf{p}$ and $\mathbf{q}$ represent the net active and reactive power injections, computed from fixed consumption and controllable generation, as previously defined.

The learned model $f^{\mathrm{NN}}(\cdot)$ provides a differentiable, structured mapping from net injections to voltages. Constructed with ReLU-based neurons, $f^{\mathrm{NN}}(\cdot)$ is continuous but piecewise-linear. Therefore, the constraint (3b) introduces combinatorial structure through binary activation patterns. As shown in [6], this structure can be reformulated into mixed-integer linear equalities, which makes the overall problem a mixed-integer quadratic program (MIQP).

In this formulation, the controllable reactive power $\mathbf{q}^g$ and resulting voltages $\mathbf{v}$ are optimization variables, while $\mathbf{p}^c, \mathbf{q}^c$ remain fixed. The PWL-NN surrogate thus enables embedding power flow physics into the optimization in a compact and computationally tractable manner, maintaining consistency and interoperability.

4. VVC on `NEO-Grid` via DEQ-based implicit learning

We now formally formulate the VVC rule design problem, building on the framework introduced in [3; 9], and present its solution leveraging the `NEO-Grid` neural voltage model. The solution employs a **deep-equilibrium model (DEQ)** [10; 11] for memory-efficient learning of optimal parameters under recursive system dynamics, as detailed in the following section.

As prescribed by the IEEE 1547 standard [13], smart inverter VVC rules typically follow a **piecewise linear (PWL)** function that maps the local voltage measurement v_i at an inverter node $i \in \mathcal{D}$ to its reactive power output q_i^g. Figure 2 depicts this PWL mapping.

This rule is parameterized by four interpretable quantities: nominal voltage setpoint $\bar{v}$, deadband half-width δ, ramp width σ, and maximum reactive power magnitude $\bar{q}$. The slope of the ramp regions is given by:

$$\alpha = \frac{\bar{q}}{\sigma - \delta}, \tag{4}$$

where $\sigma - \delta$ is the ramp region outside the deadband.

The IEEE 1547 standard also imposes the following bounds on these parameters to ensure safe and interoperable operation:

$$0.95 \le \bar{v} \le 1.05 \tag{5a}$$

$$0 \le \delta \le 0.03 \tag{5b}$$

$$\delta + 0.02 \le \sigma \le 0.18 \tag{5c}$$

$$0 \le \bar{q} \le \hat{q}. \tag{5d}$$

We explicitly define the VVC control rule as the mapping:

$$\mathbf{q}^g = g(\mathbf{v}; \phi)$$

where $g(\cdot;\boldsymbol{\phi})$ is the PWL function applied at each inverter node with parameters $\boldsymbol{\phi} := \{\boldsymbol{\phi}_i\}_{i\in\mathcal{D}}$.

The VVC operates in a closed loop: each inverter measures its local voltage v_i, computes $q_i^g = g(v_i;\boldsymbol{\phi}_i)$, and the updated $\mathbf{q}^g$ is used in the power flow equations to compute a new voltage profile $\mathbf{v}$. This process iterates until the system converges to a steady-state voltage profile $\mathbf{v}^*$, which is typically achieved quickly due to the fast dynamics of smart inverters [5].

The **VVC rule design problem** — as originally formalized in [3; 9] — is to determine the optimal parameter vector $\boldsymbol{\phi}$ such that the resulting $\mathbf{v}^*$ is as close as possible to the flat profile at 1.0 pu, while satisfying the IEEE standard bounds:

$$\min_{\boldsymbol{\phi}} \quad \|\mathbf{v}^* - \mathbf{1}\|_2^2 \tag{6a}$$

$$\text{s.t.} \quad \mathbf{v}^* = f\big(\mathbf{p}^g - \mathbf{p}^c, \mathbf{q}^g \leftarrow g(\mathbf{v}^*;\boldsymbol{\phi}) - \mathbf{q}^c\big) \tag{6b}$$

$$\text{IEEE standard bounds in (5)}$$

Here, $f(\cdot)$ denotes the power flow mapping (or its neural surrogate f^{NN}).

This formulation highlights the coupled nature of voltages and reactive power through $g(\cdot)$ and $f(\cdot)$, which results in a non-convex optimization problem with both continuous and combinatorial elements. As such, it resembles a mixed-integer nonlinear program (MINLP) and poses computational challenges in large-scale settings.

Remark 3 (On Stability of the Closed-Loop Dynamics)**.** *Stability constraints can be incorporated into the VVC rule design to promote convergence of the closed-loop dynamics. Although a closed-form stability criterion for the nonlinear system is not readily available, adopting the linear-system constraints from [3] is still meaningful in practice, since the actual distribution network does not exactly follow either the linearized or the neural surrogate model.*

In the linearized VVC formulation [3], stability is typically enforced via spectral-norm conditions such as

$$\|\mathrm{dg}(\boldsymbol{\alpha})\mathbf{X}\|_2 \le 1 - \epsilon,$$

or their polytopic relaxation

$$\mathbf{X}\boldsymbol{\alpha} \le (1-\epsilon)\mathbf{1}, \quad \alpha_n \le \frac{1-\epsilon}{\sum_{m\in\mathcal{N}} X_{nm}}, \ \forall n \in \mathcal{N}.$$

These constraints ensure that the fixed-point iteration converges in the linear setting and can be similarly adopted here to encourage stable closed-loop behavior.

4.1 DEQ for `NEO-Grid` Embedded VVC Design

We represent the VVC control rule $g(\cdot;\boldsymbol{\phi})$ as a structured ReLU neural network with fixed weights and parameterized breakpoints, following [3; 9]. This neural formulation translates the classical PWL logic into a compact, differentiable architecture suitable for gradient-based optimization. Figure 3 illustrates how the local voltage v_i is transformed through affine operations and ReLU activations into the reactive power output q_i^g, while preserving physical interpretability.

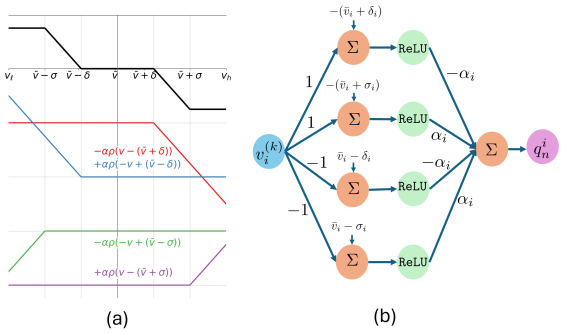

Figure 3: Neural implementation of the VVC rule $g(v_i;\boldsymbol{\phi})$ as a structured ReLU network, mapping voltage v_i to reactive power q_i^g through affine transformations and ReLU activations.

Once the control rule is defined, it is coupled with the network voltage response model. Since reactive power injections $\mathbf{q}$ influence bus voltages $\mathbf{v}$, which in turn determine $\mathbf{q}$ through the control rule, this forms a closed-loop dynamic captured by the iterative updates:

$$\mathbf{q}^{g(k)} = g(\mathbf{v}^{(k)};\boldsymbol{\phi}) \tag{7a}$$

$$\mathbf{v}^{(k+1)} = f^{\text{NN}}\big(\mathbf{p}^g - \mathbf{p}^c, \mathbf{q}^{g(k)} - \mathbf{q}^c\big) \tag{7b}$$

Here, $\mathbf{v}^{(k)}$ denotes the voltage profile at iteration k, and $\mathbf{q}^{(k)}$ is the reactive power vector computed from the VVC rule applied to $\mathbf{v}^{(k)}$. The function $f^{\text{NN}}(\cdot)$ represents the voltage response model, learned as a surrogate for the nonlinear power flow equations.

In prior work [3; 9], this closed-loop recursion is implemented as a recurrent neural network (RNN), where the sequence $\{\mathbf{v}^{(k)}\}$ is unrolled and gradients are back-propagated through all intermediate states. However, unrolling suffers from vanishing gradients, high memory usage, and sensitivity to initialization — making training inefficient and unstable in practice.

To overcome these limitations, we leverage the Deep Equilibrium Model (DEQ) framework [10; 11], which directly solves for the steady-state voltage profile $\mathbf{v}^*$ as the fixed point of the closed-loop dynamics:

$$\mathbf{v}^* = f^{\text{NN}}\big(\mathbf{p}^g - \mathbf{p}^c, g(\mathbf{v}^*;\boldsymbol{\phi}) - \mathbf{q}^c\big) \tag{8}$$

Instead of unrolling the recursion, the DEQ formulation employs root-finding algorithms (e.g., Anderson acceleration) to compute $\mathbf{v}^*$ efficiently. Once the fixed point

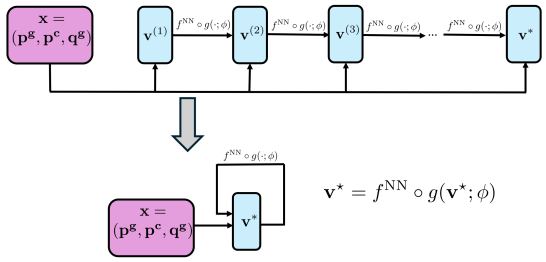

Figure 4: Comparison of iterative unrolling and DEQ for volt-var dynamics. Traditional unrolling repeatedly applies $f^{\mathrm{NN}} \circ g$ until convergence, consuming memory and computation. DEQ directly solves the fixed-point $\mathbf{v}^*$ using root-finding and enables efficient implicit differentiation.

is found, gradients of the loss with respect to $\boldsymbol{\phi}$ can be computed via implicit differentiation, which bypasses the need to store intermediate iterates and improves both memory and computational efficiency.

We formalize the fixed-point problem as solving:

$$\mathbf{v}^* = \mathcal{F}_{\boldsymbol{\phi}}(\mathbf{v}^*, \mathbf{p}^g - \mathbf{p}^c, \mathbf{q}^c) \tag{9}$$

where $\mathcal{F}_{\boldsymbol{\phi}}(\cdot) := f^{\mathrm{NN}}\big(\mathbf{p}^g - \mathbf{p}^c, g(\mathbf{v}^*;\boldsymbol{\phi}) + \mathbf{q}^c\big)$ combines the VVC rule $g(\cdot;\boldsymbol{\phi})$ and the learned power flow surrogate f^{NN}.

This root-finding problem resembles solving the power flow equations at steady-state and can be efficiently handled using Anderson acceleration or similar solvers. Figure 4 illustrates the distinction between iterative unrolling and the DEQ approach.

Implicit Differentiation for Fixed-Point Models Once the equilibrium voltage $\mathbf{v}^*$ is computed, gradients with respect to the control parameters $\boldsymbol{\phi}$ can be obtained without backpropagating through all intermediate iterates. Instead, we apply the **implicit function theorem**, which gives:

$$\frac{d\mathcal{L}}{d\boldsymbol{\phi}} = -\left(\mathbf{I} - \frac{\partial \mathcal{F}_{\boldsymbol{\phi}}}{\partial \mathbf{v}}\right)^{-1} \cdot \frac{\partial \mathcal{F}_{\boldsymbol{\phi}}}{\partial \boldsymbol{\phi}} \cdot \nabla_{\mathbf{v}^*}\mathcal{L} \tag{10}$$

where $\mathcal{F}_{\boldsymbol{\phi}}(\mathbf{v})$ denotes the closed-loop mapping from voltages to voltages.

The loss is defined as:

$$\mathcal{L}(\boldsymbol{\phi}) = \|\mathbf{v}^* - \mathbf{1}\|_2^2 \tag{11}$$

This promotes voltage profiles close to nominal and enables physics-informed learning without needing ground-truth voltage or reactive power labels.

Training Pipeline The control parameters $\boldsymbol{\phi}$ — a subset of $(\alpha, \sigma, \delta, \bar{v})$ — are optimized via gradient descent.

Algorithm 1: Training Optimal Volt-Var Control Rule via DEQ

Input: Loading scenarios $\{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i = (\mathbf{p}_i^c, \mathbf{p}_i^g, \mathbf{q}_i^c)$
Output: Optimized control rule parameters $\boldsymbol{\phi}$
Initialize control rule parameters $\boldsymbol{\phi}$
foreach *scenario* $\mathbf{x}_i$ **do**
  Solve for fixed point voltage $\mathbf{v}^*$:
  $$\mathbf{v}^* = \mathcal{F}_{\boldsymbol{\phi}}(\mathbf{v}^*; \mathbf{x}_i)$$
  Compute loss:
  $$\mathcal{L}_i = \|\mathbf{v}^* - \mathbf{1}\|_2^2$$
  Compute gradient $\nabla_{\boldsymbol{\phi}}\mathcal{L}_i$ using Equation (10)
  Update $\boldsymbol{\phi}$ via gradient descent
return $\boldsymbol{\phi}$

For each sampled loading scenario $\mathbf{x}_i = (\mathbf{p}_i^c, \mathbf{p}_i^g, \mathbf{q}_i^c)$, we solve the fixed-point equation for $\mathbf{v}^*$, compute the loss, and update $\boldsymbol{\phi}$ using the implicit gradient from Equation (10). The full training procedure is summarized in Algorithm 1.

Remark 4 (On Stability)**.** *The proposed DEQ-based VVC design is inherently an equilibrium problem: convergence of the fixed-point iteration implies stable closed-loop behavior. In practice, the convergence (and hence stability) of the iterative solver depends on the step size and initialization. We observe that using smaller step sizes and well-chosen initializations promotes convergence to a stable voltage profile. Future work will explicitly incorporate stability considerations into the training and deployment of the control rule.*

5. Numerical Results

Experimental Setup. We evaluate all models and control strategies on the IEEE 33-bus radial distribution network [4], which consists of $N = 32$ load buses and one substation. This network is widely used in distribution system research due to its realistic topology and moderate complexity.

All simulations and experiments were conducted on a personal laptop (Dell Inspiron 16 Plus 7640) running Windows 11 Home (Build 26100), equipped with an Intel CPU (Intel64 Family 6 Model 170, ∼1.4 GHz), 32 GB of RAM, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB of VRAM.

Neural network models were implemented and trained using PyTorch. Data generation and power

flow simulations were performed using the Pandapower library. All experiments and results are fully reproducible; the source code and datasets will be made publicly available on GitHub upon publication.

Experimental Datasets. We construct two distinct datasets to support the learning-based modeling and control evaluations. All data is generated using the `Pandapower` toolbox, using a fixed nominal operating point for the IEEE 33-bus network (see Section 3). At this nominal point, all active and reactive power values are interpreted as consumption. All quantities are expressed in per-unit (p.u.) on a 1 MVA base.

The **first dataset** is used to train the neural network power flow approximation model, as described in Section 3. Each data point consists of a 64-dimensional input vector, formed by concatenating the net active and reactive power at all $N = 32$ buses: $[\mathbf{p}; \mathbf{q}] \in \mathbb{R}^{64}$. The output is the corresponding bus voltage vector $\mathbf{v} \in \mathbb{R}^{32}$, computed via an AC power flow simulation. Thus, each row of the dataset contains 96 columns (64 inputs + 32 voltage outputs).

This dataset contains 20,000 samples, with an 80/20 train-test split. The net power demands are generated by randomly perturbing the nominal active and reactive power values within $\pm 10\%$. Importantly, to ensure that the learned voltage model generalizes to scenarios with reactive power compensation (i.e., $\mathbf{q^g} < 0$), we inject an additional layer of perturbation to the reactive power entries: each bus's net $\mathbf{q}$ is randomly perturbed within the range $[-0.8, +0.2]$ p.u. This enables the NN model to learn accurate voltage responses in settings where reactive generation may be present. This additional perturbation applies to both training and test splits of the dataset. Once trained and validated, the neural network no longer uses this dataset in downstream control tasks.

The **second dataset** is used exclusively for evaluating and training VVO and VVC schemes. It contains 100 samples, generated with the same $\pm 10\%$ perturbation to nominal active and reactive power demands, but without any artificial correction to the reactive power. As such, the reactive power profile remains consistent with realistic consumption scenarios, and will instead be adjusted by the VVO and VVC algorithms. As before, each sample yields a total of 96 columns via power flow simulation.

The dataset is again split 80/20. The first 80 samples are used to train the VVC control rule parameters (see Section 4), while the final 20 samples are used to evaluate both VVO and VVC performance under unseen loading conditions. The VVO method, being optimization-based, does not require a training phase.

Table 1: **Test set voltage prediction error (MSE) for different power flow approximation models.** The neural network significantly outperforms both LinDistFlow and linear regression, validating the importance of nonlinear modeling in accurately capturing power flow behavior.

Model	Voltage MSE (Test Set)
LinDistFlow	1.95×10^{-4}
Linear Regression (LS)	3.16×10^{-5}
NEO-Grid (NN)	$\mathbf{1.81 \times 10^{-6}}$

5.1 Power Flow Approximation Performance.

To evaluate the accuracy of learning-based voltage modeling, we train a neural network to approximate the mapping from bus-level power consumption to voltage magnitudes. Specifically, the model takes as input the net real and reactive power demands at all $N = 32$ buses, forming a 64-dimensional input vector $[\mathbf{p}; \mathbf{q}]$, and outputs the 32-dimensional voltage vector $\mathbf{v}$. The training is performed on the first dataset described previously, which covers a wide range of operating conditions via $\pm 10\%$ perturbations and reactive compensation ranging from -0.8 to $+0.2$ p.u.

Baseline Models. We benchmark the proposed neural network model against two standard alternatives. First, we use the **LinDistFlow** model, a physics-inspired linear approximation derived from first principles, which requires no data-driven training. Second, we implement a **linear regression** model, equivalent to a neural network with no hidden layer. This serves as a data-driven linear baseline trained on the same perturbed dataset, yielding a single affine mapping from power injections to voltages. These comparisons help isolate the benefits of nonlinearity in neural modeling.

Neural Network Setup. The neural network used for voltage prediction consists of one hidden layer with $K = 64$ ReLU units. The model is trained using PyTorch and optimized using the Adam optimizer with a learning rate of 10^{-4}, over 50,000 epochs with batch size 1. Input and output data are scaled to $[0, 1]$ using a min-max normalization fitted to the data. All training is conducted on a Dell Inspiron 16 Plus 7640 laptop using an NVIDIA GeForce RTX 4060 Laptop GPU (8 GB VRAM), with PyTorch automatically leveraging GPU acceleration.

Results. We evaluate each model on the 20% test split (4,000 samples), computing the mean squared error (MSE) between predicted and true voltages over all nodes. As shown in Table 1, the neural network achieves an MSE of 1.81×10^{-6}, which is an order of magnitude

better than the linear regression model (3.16×10^{-5}) and over 100x better than the LinDistFlow approximation (1.95×10^{-4}).

These results confirm that learning a nonlinear mapping using a hidden layer with ReLU activation yields a much more expressive and accurate model of the power flow dynamics. While the LinDistFlow model is analytically convenient and the linear regression model adapts to data, both fail to capture the nonlinear coupling between nodal powers and voltages that arise due to network topology and line impedances. The neural network, in contrast, learns to accurately account for these effects, making it a more suitable surrogate for downstream optimization and control.

5.2 VVO Test Results.

Optimization Setup. We solve the volt-var optimization problem formulated in (2) for the 20 testing samples in the second dataset. The optimization is implemented using the `Pyomo` framework [16], with the neural network power flow surrogate embedded via the `OMLT` toolkit [17]. We enforce inverter reactive power constraints by bounding the decision variables $\mathbf{q}^g$ within $[-0.6, 0.1]$ p.u. at designated DER nodes.

To improve tractability, we approximate the quadratic objective $\|\mathbf{v} - \mathbf{1}\|_2^2$ using a linear surrogate based on the ℓ_1 norm: $\|\mathbf{v} - \mathbf{1}\|_1$. This relaxation enables solving the neural VVO problem as a mixed-integer linear program (MILP) rather than a computationally expensive MIQP. The same ℓ_1 formulation is used across all models to ensure comparability. All problems are solved using the open-source `CBC` solver [18], which supports mixed-integer linear optimization.

Results and Analysis. We evaluate each method by computing voltage deviation from the nominal 1.0 p.u. profile across all buses and test scenarios. We report: (i) the average absolute deviation across all bus-voltage values; and (ii) the percentage of voltage entries with deviation exceeding 1%, 3%, and 5%. These statistics are computed over $20 \times 32 = 640$ voltage entries. Table 2 summarizes the results.

The results clearly show that all three optimization-based methods significantly improve voltage regulation compared to the no-correction baseline, which yields widespread violations. Among the methods, the NEO-Grid neural network model consistently achieves the best performance. It not only delivers the smallest average absolute voltage deviation (0.94%) but also substantially reduces the incidence of large voltage violations: only 5.47% of buses exceed a 3% deviation, compared to over 10% for linear regression and more than 11% for LinDistFlow. No model exceeds the 5% threshold, confirming the effectiveness of all methods in enforcing standard voltage limits. These findings emphasize the value of nonlinear modeling via neural networks in distribution system control, especially in minimizing extreme deviations that could compromise system reliability.

5.3 Optimal VVC Test Results.

DEQ-Based VVC Implementation. To implement volt-var control within the `NEO-Grid` framework, we adopt the Deep Equilibrium Model (DEQ) architecture as described in [19]. We design a minimalist DEQ structure with a single equilibrium layer responsible for modeling the closed-loop voltage dynamics induced by the control rules. No additional feedforward layers are used. The DEQ layer maps an initial voltage estimate $\mathbf{v}^{(0)}$ and an exogenous input $\mathbf{x} = (\mathbf{p}^c, \mathbf{p}^g, \mathbf{q}^c)$ to a fixed-point solution $\mathbf{v}^*$, which satisfies the condition:

$$\mathbf{v}^* = \mathcal{F}_{\boldsymbol{\phi}}(\mathbf{v}^*, \mathbf{p}, \mathbf{q}^c)$$

DEQ Training Details. Training is conducted on the first 80 samples of the second dataset, using batch training over mini-batches of size 16. The total number of training epochs is 500, with a learning rate of 10^{-3}. The training process follows the batch-style DEQ optimization scheme introduced in [19], where equilibrium is approximated per batch using Anderson acceleration. To enforce the feasibility of learned control rules with respect to the IEEE 1547 standard (see Eq. (5)), we apply **projected gradient descent** at each training step. This ensures that parameter updates are immediately projected back onto the feasible region, a process that is readily incorporated into the PyTorch optimization pipeline.

Results and Analysis. Table 3 summarizes the performance of various volt-var control strategies across 20 test scenarios. The DEQ-trained NEO-Grid controller yields the **lowest average deviation of 3.63%**—outperforming all alternatives—and successfully eliminates all voltage entries with deviations at and beyond 7%. This is in stark contrast to the other optimization-based models (LinDistFlow and Linear Regression), which still experience notable large deviations beyond 5% and 7%. Notably, while LinDistFlow and LS reduce overall error relative to the initial or uncontrolled profiles, they fail to match the robustness and consistency achieved by the learned DEQ controller. The no-correction baseline ($\mathbf{q}^g = \mathbf{0}$) performs the worst across all metrics, with more than 65% of voltage entries deviating by over 5%, confirming the necessity of reactive control. Overall, these results underscore the ability of equilibrium-based neural control design to deliver high-accuracy regulation under diverse load and generation conditions.

Table 2: **Voltage deviation statistics for different volt-var optimization models across 20 test scenarios.** The first column shows the average absolute deviation from 1.0 p.u., while the remaining columns report the percentage of voltage entries exceeding 1%, 3%, and 5% deviation thresholds, respectively. All optimization-based approaches significantly outperform the no-correction baseline. The neural network model (NEO-Grid) achieves the best performance across all metrics, including the lowest average error and the smallest rates of large violations.

Model	**Avg Deviation**	$> 1\%$	$> 3\%$	$> 5\%$
NEO-Grid (NN)	**0.94%**	**28.13%**	**5.47%**	**0.00%**
Linear Regression (LS)	0.97%	31.25%	10.78%	0.00%
LinDistFlow	1.04%	30.47%	11.72%	0.00%
No Correction ($\mathbf{q}^g = \mathbf{0}$)	5.32%	84.38%	71.88%	65.31%

Table 3: **Voltage regulation performance of different VVC models under 20 test scenarios.** The first column reports the average absolute deviation from the nominal 1.0 p.u. voltage profile. The next two columns show the percentage of bus voltages with less than 5% and 7% deviation, respectively. The DEQ-based NEO-Grid model achieves the lowest average deviation and eliminates all violations at and beyond 7%, highlighting its superior ability to learn effective volt-var control policies. All optimization-based strategies improve substantially over the initial and no-correction baselines.

Model	**Avg Deviation**	$>$ **5%**	$>$ **7%**
NEO-Grid (NN)	**3.63%**	**38.59%**	**0.00%**
LinDistFlow	4.66%	53.12%	28.12%
Linear Regression (LS)	4.55%	50.00%	26.56%
Initial Weights (No Optimization)	4.76%	53.12%	18.75%
No Correction ($\mathbf{q}^g = \mathbf{0}$)	5.32%	65.31%	43.59%

VVO versus VVC Performance. It is worth noting that the VVO formulation achieves consistently better voltage regulation than the optimal VVC methods across all metrics. This is expected: in VVO, the inverter reactive power injections $\mathbf{q}^g$ are treated as free decision variables constrained only by physical bounds, allowing the optimization to select values that minimize deviation directly. In contrast, VVC policies are restricted to follow a learned piecewise-linear control rule with a fixed structure—parameterized by $(\bar{v}, \sigma, \delta, \bar{q})$—that maps voltage measurements to reactive power responses. As such, even under optimal parameter tuning, the VVC framework cannot fully match the flexibility and performance of the unconstrained VVO problem. This highlights the inherent trade-off between policy interpretability and closed-loop control fidelity.

6. Conclusions

This paper presented `NEO-Grid`, a unified learning-based framework for volt-var optimization and control in distribution systems. By integrating nonlinear neural surrogates for power flow and equilibrium-based control via deep equilibrium models (DEQs), our approach captures closed-loop voltage dynamics while remaining tractable for both optimization and learning. Experiments on the IEEE 33-bus network show that NEO-Grid consistently outperforms traditional linear and heuristic methods in voltage deviation and constraint satisfaction. Future work includes extending the framework to multi-phase systems, stochastic inputs, and real-time deployment.

References

[1] A. Srivastava, J. Zhao, H. Zhu, F. Ding, S. Lei, I. Zografopoulos, R. Haider, S. Vahedi, W. Wang, G. Valverde *et al.*, "Distribution system behind-the-meter ders: Estimation, uncertainty quantification, and control," *IEEE Transactions on Power Systems*, 2024.

[2] Y. Chen, Y. Shi, and B. Zhang, "Data-driven optimal voltage regulation using input convex neural networks," *Electric Power Systems Research*, vol. 189, p. 106741, 2020.

[3] S. Gupta, V. Kekatos, and S. Chatzivasileiadis, "Optimal design of volt/var control rules of inverters using deep learning," *IEEE Transactions on Smart Grid*, 2024.

[4] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power delivery*, vol. 4, no. 2, pp. 1401–1407, 2002.

[5] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: Optimality and stability analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3794–3803, 2016.

[6] Y.-h. Cho and H. Zhu, "Data-driven modeling of linearizable power flow for large-scale grid topology optimization," *arXiv preprint arXiv:2409.13956*, 2024.

[7] J. Chen, W. Wu, and L. A. Roald, "Data-driven piecewise linearization for distribution three-phase stochastic power flow," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1035–1048, 2022.

[8] G. Chen, H. Zhang, and Y. Song, "Efficient constraint learning for data-driven active distribution network operation," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1472–1484, 2024.

[9] S. Gupta, A. Mehrizi-Sani, S. Chatzivasileiadis, and V. Kekatos, "Deep learning for scalable optimal design of incremental volt/var control rules," *IEEE Control Systems Letters*, vol. 7, pp. 1957–1962, 2023.

[10] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," *Advances in neural information processing systems*, vol. 32, 2019.

[11] S. Bai, V. Koltun, and J. Z. Kolter, "Multiscale deep equilibrium models," *Advances in neural information processing systems*, vol. 33, pp. 5238–5250, 2020.

[12] M. E. Baran and F. F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Transactions on power Delivery*, vol. 4, no. 1, pp. 725–734, 2002.

[13] "Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pp. 1–138, 2018.

[14] A. Kody, S. Chevalier, S. Chatzivasileiadis, and D. Molzahn, "Modeling the ac power flow equations with optimally compact neural networks: Application to unit commitment," *Electric Power Systems Research*, vol. 213, p. 108282, 2022.

[15] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive ac-opf solutions," *IEEE Transactions on Power Systems*, vol. 38, no. 6, pp. 5660–5670, 2022.

[16] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, J. D. Siirola *et al.*, *Pyomo-optimization modeling in python*. Springer, 2017, vol. 67.

[17] F. Ceccon, J. Jalving, J. Haddad, A. Thebelt, C. Tsay, C. D. Laird, and R. Misener, "Omlt: Optimization & machine learning toolkit," *Journal of Machine Learning Research*, vol. 23, no. 349, pp. 1–8, 2022.

[18] J. Forrest, "CBC (Coin-or branch and cut)," https://github.com/coin-or/Cbc, 2023, version 2.10.10.

[19] Deep Implicit Layers Tutorial Team, "Chapter 4: Deep equilibrium models," https://implicit-layers-tutorial.org/deep_equilibrium_models/, accessed: 2025-06-15.