

Comparing UDEs and BNODEs for Microgrid Dynamics: Structure-Preserving Models with Calibrated Uncertainty

Samarth Sharma Prathamesh Dinesh Joshi
Raj Abhijit Dandekar Rajat Dandekar Sreedhat Panat

Antimatters Labs Vizura AI Labs
Massachusetts Institute of Technology (prior) Purdue University (prior)
samarth@antimatterslabs.com,
{prathamesh, raj, rajatdandekar, sreedhat}@vizura.com

ABSTRACT

Microgrids, localized electrical grids that can operate independently or connected to the main power system, are emerging as critical infrastructure for grid resilience and renewable energy integration. However, high renewable penetration creates new challenges: these systems must regulate frequency and power under fast, nonlinear dynamics and mode switches, making reliable, real-time models essential for control and planning. Classical differential-algebraic simulators capture detailed physics but suffer computational bottlenecks and parameter sensitivity, while purely black-box learners achieve fast inference but violate physical constraints and fail under distribution shift. Scientific Machine Learning addresses this trade-off by fusing mechanistic structure with learnable components and calibrated uncertainty. We address this trade-off by comparing two neural ODE approaches: Universal Differential Equations that learn residual $f_\theta(P_{\text{gen}})$ inside mechanistic models, and Bayesian Neural ODEs that treat all dynamics probabilistically. For calibration’s comparison, we apply post-hoc calibration to both methods: global affine mapping with conformal prediction for UDEs, and variance scaling for BNODEs. On 10 held-out scenarios, UDEs achieve physics-comparable accuracy while producing interpretable quintic residuals ($R^2 = 0.9998$). BNODEs provide principled uncertainty quantification with calibration improving 90% coverage from 0.866 to 0.893. These calibrated approaches, maintaining millisecond inference, offer practical alternatives to black-box methods, with accuracy and interpretability for microgrid control.

1 INTRODUCTION

Renewable-rich microgrids integrate inverter-based distributed energy resources (DERs) to enhance resilience and sustainability, but their dynamics pose fundamen-

tal modeling challenges beyond traditional power system analysis [9, 23, 36]. Unlike synchronous generators with well-characterized swing dynamics, inverter-based resources exhibit fast switching behaviors, complex control interactions, and mode-dependent responses that defy conventional DAE-based models [12, 21, 14]. Grid-forming inverters introduce additional nonlinearities through droop control mechanisms that dynamically couple active power with frequency and reactive power with voltage [27, 34]. Modern microgrid operation requires models capturing these dynamics across multiple temporal scales while remaining computationally tractable for real-time control and extrapolable beyond training conditions [28].

Traditional approaches face a fundamental dilemma. High-fidelity DAE simulators capture detailed device physics but require extensive parameter identification, exhibit computational bottlenecks, and often lack critical parameters in practical deployments [27, 8]. Numerical challenges become acute with stiff systems exhibiting multiple time scales, requiring specialized methods for stable integration [13, 4]. Conversely, purely data-driven methods efficiently fit observed behaviors but typically ignore physical constraints, leading to conservation law violations and poor extrapolation [16, 6].

PINNs embed differential equations as soft constraints in training objectives [33, 16], but recent analysis reveals systematic failure modes where PINNs converge to trivial solutions, particularly with sparse data or competing loss terms [19, 41, 42]. These challenges intensify for stiff differential equations common in power systems, where PINNs struggle with stability and convergence [15, 19]. Spectral bias causes PINNs to favor smooth solutions even when true dynamics contain sharp transients essential for power system modeling [32, 38, 40, 37].

Neural ODEs represent continuous-time dynamics, enabling irregular sampling, memory-efficient training

via adjoint methods, and natural constraint incorporation [5, 17]. Universal Differential Equations address structural limitations by embedding neural networks within mechanistic models, preserving known physics while learning unknown components [31]. This hybrid approach maintains interpretability while capturing complex nonlinear effects [2]. Recent advances in stiff ODE solvers make UDEs practical for power systems with vastly different time scales [29, 30]. Bayesian Neural ODEs extend this framework with probability distributions over parameters, enabling principled uncertainty quantification that distinguishes epistemic from aleatoric uncertainty [7]. The Bayesian framework naturally accommodates parameter uncertainty and provides principled model selection [10].

Uncertainty estimates must be carefully calibrated to achieve nominal coverage rates [20, 11]. Recent advances demonstrate that calibration is especially critical for extrapolation in operator learning [22, 25, 26]. Proper scoring rules provide frameworks for evaluating probabilistic forecasts [11, 3]. Conformal prediction offers distribution-free uncertainty with finite-sample guarantees, though physics-grounded approaches often provide better-calibrated estimates when extrapolating [35, 1, 26, 39]. Operator learning methods including FNOs and DeepONets provide alternative paradigms for learning solution operators rather than individual instances [22, 25, 18], offering faster inference for repeated evaluations but often exhibiting poor uncertainty calibration under distribution shift [26, 24].

This comparison addresses a critical gap in scientific ML for power systems. While PINNs have been explored extensively [33, 16], their failure modes with stiff systems [15, 19] make them unsuitable for fast-timescale microgrid dynamics. Our work demonstrates that hybrid approaches can match physics-based accuracy while adding interpretability (UDEs) or calibrated uncertainty (BNODEs), providing practical alternatives for real-time control and planning under renewable uncertainty.

We make the following contributions: (1) First systematic comparison of UDEs versus BNODEs on identical dynamical systems, with evaluation spanning point accuracy, probabilistic calibration, symbolic interpretability, and computational efficiency; (2) Demonstration that UDEs preserve physics-comparable accuracy while enabling post-hoc symbolic distillation—a quintic in P_{gen} with $R^2 = 0.9998$ extracted from 11 neural parameters; (3) Characterization of computational efficiency showing UDEs achieve $2.8\times$ faster training than BNODEs on our dataset due to MCMC inference overhead; (4) Deployment guidance providing practical method selection framework based on operational

Table 1: Model variables and parameters (per-unit unless noted).

Symbol	Description	Range/Units
x_1	Storage state-of-charge	[0, 1]
x_2	Frequency/power deviation	[-1, 1]
$\eta_{\text{in}}/\eta_{\text{out}}$	Charge/discharge efficiency	[0.85, 0.95]
α	Damping coefficient	[0.1, 0.5]
β	Power-frequency coupling	[0.8, 1.2]
γ	Storage-grid coupling	[0.025, 0.049]
u	Control input	0 (open-loop)
$d(t)$	Disturbance/self-discharge	time-varying
$P_{\text{gen}}/P_{\text{load}}$	Generation/load power	[0, 1]

requirements.

2 METHODS

2.1 Two-State Microgrid Model

We adopt a reduced-order model that captures essential microgrid dynamics while remaining tractable for systematic analysis. The model represents storage-frequency interactions inspired by droop control and swing dynamics:

$$\frac{dx_1}{dt} = \eta_{\text{in}} \cdot u \cdot \mathbb{K}_{\{u>0\}} - \frac{1}{\eta_{\text{out}}} \cdot u \cdot \mathbb{K}_{\{u<0\}} - d(t), \quad [28, 9] \quad (1)$$

$$\frac{dx_2}{dt} = -\alpha x_2 + \beta \cdot P_{\text{gen}} - \beta \cdot P_{\text{load}} + \gamma x_1, \quad [34, 12, 21] \quad (2)$$

where variables and parameters are defined in Table 1. We evaluate open-loop response with $u = 0$ throughout; the control term remains in Eq. (1) for generality.

This model captures key phenomena: storage dynamics with efficiency losses (Eq. 1) and frequency-power coupling with damping and storage interaction (Eq. 2). While simplified, it exhibits essential microgrid characteristics including energy conservation, stability properties, and nonlinear responses that challenge purely data-driven approaches.

2.2 Universal Differential Equation Approach

The UDE framework preserves essential physical structure by embedding neural components within mechanistic models. We preserve the well-understood storage dynamics (Eq. 1) while learning corrections to uncertain generation-frequency coupling:

$$\frac{dx_1}{dt} = \eta_{\text{in}} \cdot u \cdot \mathbb{K}_{\{u>0\}} - \frac{1}{\eta_{\text{out}}} \cdot u \cdot \mathbb{K}_{\{u<0\}} - d(t), \quad (3)$$

$$\frac{dx_2}{dt} = -\alpha x_2 + f_{\theta}(P_{\text{gen}}) - \beta \cdot P_{\text{load}} + \gamma x_1. \quad (4)$$

Table 2: UDE hyperparameter search space and optimal configuration, searching performed over 5,760 configurations.

Hyperparameter	Search Space	Optimal
Hidden width	{3, 4, 5, 6, 8, 10}	3
L2 reg. λ	$\{10^{-6}, 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$	10^{-6}
Learning rate	$\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$	10^{-3}
ODE solver reitot	$\{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$	10^{-7}
Random seeds	{0, 1, ..., 9}	7

Storage dynamics are well-characterized by conservation laws, while generation-frequency coupling involves complex inverter dynamics difficult to model precisely. By learning only the uncertain coupling term $\beta P_{\text{gen}} \rightarrow f_{\theta}(P_{\text{gen}})$, the UDE maintains physical interpretability while capturing missing nonlinearities.

2.2.1 Neural architecture and training

The residual function $f_{\theta}(P_{\text{gen}})$ is a single-hidden-layer tanh MLP with linear readout:

$$f_{\theta}(P_{\text{gen}}) = \mathbf{w}_2^{\top} \tanh(\mathbf{W}_1 P_{\text{gen}} + \mathbf{b}_1) + b_2, \quad (5)$$

where $\mathbf{W}_1, \mathbf{b}_1, \mathbf{w}_2 \in \mathbb{R}^3$ and $b_2 \in \mathbb{R}$. Together with the physics parameter β , this yields approximately 11 trainable parameters. We selected width-3 through systematic validation: widths 4-6 achieved similar RMSE (0.247-0.249) but degraded symbolic extraction fidelity (R^2 dropping to 0.985-0.992 for quintic fits), while width-10 introduced overfitting (validation loss increased 15% despite lower training loss). We validated this 1D design by comparing against a five-input variant $r(P_{\text{gen}}, P_{\text{load}}, x_1, x_2, t)$; results were qualitatively similar ($\Delta \text{RMSE} < 0.003$), confirming that generation-frequency coupling is primarily univariate. The reported quintic represents the main effect in P_{gen} , with interaction terms empirically small (Fig. 7).

Table 2 presents the hyperparameter search space and optimal configuration identified through systematic grid search with validation-based early stopping.

We minimize a frequency-focused composite loss:

$$\mathcal{L}(\theta) = \text{RMSE}_{x_2} + 0.2 \cdot \text{RMSE}_{x_1} + 0.1 \cdot \text{MAPE}_{x_2} + \lambda \|\theta\|_2^2, \quad (6)$$

prioritizing frequency accuracy (weight 1.0) over storage (0.2) and relative errors (0.1), reflecting the operational importance of frequency stability. Regularization applies only to neural network parameters.

2.3 Bayesian Neural ODE Approach

BNODEs place probability distributions over neural ODE parameters, enabling principled uncertainty propagation. The BNODE replaces both equation compo-

Algorithm 1 UDE Training Pipeline

Require: Training data $\mathcal{D}_{\text{train}}$, validation data $\mathcal{D}_{\text{val}}$

Require: Hyperparameter space as described in Table 2

Ensure: Trained UDE model f_{θ^*} with optimal hyperparameters

```

1: Initialize: Best loss  $L^* \leftarrow \infty$ , best  $\theta^* \leftarrow \emptyset$ 
2: for each config  $(w, \lambda, \eta, \text{reitol}, s)$  in space do
3:   Initialize:  $\theta \sim \mathcal{N}(0, 0.1^2 I)$  with seed  $s$ 
4:   patience  $\leftarrow 0$ , epoch  $\leftarrow 0$ 
5:   while epoch < 1000 and patience < 50 do
6:     Solve ODE with tolerance reitol
7:     Compute loss:  $L = \text{RMSE}_{x_2} + 0.2 \cdot \text{RMSE}_{x_1} + 0.1 \cdot \text{MAPE}_{x_2} + \lambda \|\theta\|_2^2$ 
8:     Update:  $\theta \leftarrow \theta - \eta \nabla_{\theta} L$  using L-BFGS1
9:     if epoch mod 5 = 0 then
10:      Compute validation loss  $L_{\text{val}}$ ; update patience
        if  $L_{\text{val}}$  improves
11:    end if
12:    epoch  $\leftarrow$  epoch + 1
13:  end while
14:  if  $L < L^*$  then
15:     $L^* \leftarrow L$ ,  $\theta^* \leftarrow \theta$ 
16:  end if
17: end for
18: Return: Optimal model  $f_{\theta^*}$ 
19: Empirical scaling:  $O(n \log n)$  per epoch; 5,760 configs

```

nents with probabilistic neural networks:

$$\frac{dx_1}{dt} = f_{\theta_1}(x_1, x_2, u, d), \quad (7)$$

$$\frac{dx_2}{dt} = f_{\theta_2}(x_1, x_2, P_{\text{gen}}, P_{\text{load}}), \quad (8)$$

where θ_1 and θ_2 are neural network parameters with Bayesian treatment. Each f_{θ} is a one-hidden-layer tanh MLP (width 5, 4 inputs) whose output is the sum of hidden activations (no explicit output layer); inputs are $[x_1, x_2, u, d]$ for the first equation and $[x_1, x_2, P_{\text{gen}}, P_{\text{load}}]$ for the second. We prototyped standard linear readouts and stiff solvers (Rosenbrock23); outcomes were qualitatively similar.

The posterior distribution over parameters given observed trajectories $\mathbf{y}$ is:

$$p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \theta)p(\theta). \quad (9)$$

We perform inference using MCMC with the No-U-Turn Sampler (NUTS) [Algorithm 2]. Posterior predictive draws are simulated with Tsit5 (reitol = abstol = 10^{-6}); exogenous signals are linearly interpolated. We use $K = 100$ posterior draws for evaluation.

2.4 Post-Hoc Calibration

We apply post-hoc calibration to both methods to ensure uncertainty estimates achieve nominal coverage on held-out data. All coverages are computed pointwise over test timestamps for x_2 unless noted otherwise.

Algorithm 2 BNODE Bayesian Inference (MCMC/NUTS)

Require: Training data $\mathcal{D} = \{(\mathbf{u}_i, t_i)\}_{i=1}^N$
Require: $C = 4$ chains, $S = 1000$ samples, $W = 100$ warmup
Ensure: Posterior samples $\{\theta^{(c,s)}, \sigma^{(c,s)}\}$

- 1: Define priors: $\theta \sim \mathcal{N}(\mathbf{0}, 0.1^2 \mathbf{I})$; $\sigma \sim \text{TruncNormal}(\mu = 0.05, \sigma = 0.02, a = 0.01, b = 0.2)$
- 2: Define likelihood: $p(\mathbf{u}|\theta, \sigma) = \prod_{i=1}^N \text{StudentT}(\nu = 3, \mu = f_\theta(t_i), \sigma)$
- 3: **for** $c = 1$ to C **do**
- 4: Initialize: $\theta^{(c,0)} \sim p(\theta)$, $\sigma^{(c,0)} \sim p(\sigma)$
- 5: **for** $s = 1$ to $W + S$ **do**
- 6: NUTS step: build tree until U-turn or max depth 10
- 7: **if** $s \leq W$ **then**
- 8: Adapt step size and mass matrix
- 9: **end if**
- 10: Solve ODE: $\frac{d\mathbf{u}}{dt} = f_{\theta^{(c,s)}}(\mathbf{u}, t)$ (Tsit5/Rosenbrock23)
- 11: Compute $\ell^{(c,s)} = \log p(\mathbf{u}|\theta^{(c,s)}, \sigma^{(c,s)})$
- 12: **if** ODE solve fails **then**
- 13: $\ell^{(c,s)} \leftarrow -\infty$
- 14: **end if**
- 15: **end for**
- 16: **end for**
- 17: Check diagnostics: $\hat{R} < 1.05$, ESS > 400
- 18: **Return:** posterior samples after warmup
- 19: **Empirical scaling:** $O(n^2)$ per MCMC iteration

BNODE calibration via variance scaling. We compute posterior-predictive draws on validation and test sets. On validation, we fit a single global variance scale α by minimizing squared error between nominal and empirical coverage at 50% and 90% levels, using Gaussian-summary statistics (per-time mean μ and standard deviation s across draws). We search α over 80 evenly-spaced values in $[0.50, 2.00]$, computing coverage from quantiles $z_{0.75} = \Phi^{-1}(0.75) \approx 0.674$ and $z_{0.95} = \Phi^{-1}(0.95) \approx 1.645$ for intervals $\mu_i \pm z\alpha s_i$. The optimal α is then applied to test data. On our split, grid search selected $\alpha = 0.5$; other seeds sometimes yield $\alpha \approx 0.9$ with similar trade-offs.

UDE calibration via affine correction. UDE calibration applies a global affine map $\hat{y}' = a\hat{y} + b$ fit via closed-form OLS on validation predictions. Uncertainty intervals use split-conformal prediction: with validation residuals $r_i = |y_i - \hat{y}'_i|$, we form bands $[\hat{y}' \pm q_\alpha]$ where q_α is the $(1 - \alpha)$ quantile of $\{r_i\}$. On test, this calibration improves RMSE from 1.131 to 0.777 and R^2 from -0.175 to 0.445 , with coverage of 65.3% at 50% nominal and 86.8% at 90% nominal (mean widths 1.463/2.279).

Algorithm 3 BNODE Posterior Calibration

Require: Posterior samples $\{\theta^{(s)}, \sigma^{(s)}\}_{s=1}^S$ ($S = 100$)
Require: Validation data $\mathcal{D}_{\text{val}}$, test data $\mathcal{D}_{\text{test}}$
Ensure: Calibrated predictive distributions with α^*

- 1: **Generate Validation Predictions:**
- 2: **for** $i = 1$ to N_{val} **do**
- 3: **for** $s = 1$ to S **do**
- 4: $\hat{\mathbf{u}}_i^{(s)} \leftarrow \text{ODESolve}(f_{\theta^{(s)}}, t_i)$
- 5: **end for**
- 6: $\mu_i \leftarrow \frac{1}{S} \sum_{s=1}^S \hat{\mathbf{u}}_i^{(s)}$; $s_i \leftarrow \sqrt{\frac{1}{S-1} \sum_{s=1}^S (\hat{\mathbf{u}}_i^{(s)} - \mu_i)^2}$
- 7: **end for**
- 8: **Search** α over 80 evenly-spaced values in $[0.50, 2.00]$ minimizing
- 9: $(\text{cov}_{50} - 0.5)^2 + (\text{cov}_{90} - 0.9)^2$
- 10: Use $z_{0.75} = \Phi^{-1}(0.75) \approx 0.674$, $z_{0.95} = \Phi^{-1}(0.95) \approx 1.645$ for bands $\mu_i \pm z\alpha s_i$
- 11: **Apply to Test:** compute μ_j, s_j similarly; evaluate with α^*

Algorithm 4 UDE Conformal Calibration

Require: Trained UDE f_{θ^*} , validation $\mathcal{D}_{\text{val}}$, test $\mathcal{D}_{\text{test}}$
Ensure: Calibrated predictions with conformal intervals

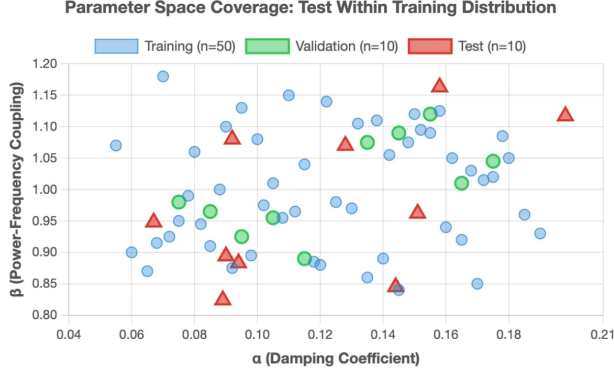
- 1: Validation: $\hat{\mathbf{u}}_i \leftarrow \text{ODESolve}(f_{\theta^*}, t_i)$; $\hat{\mathbf{y}}_i = \hat{\mathbf{u}}_i[2]$
- 2: Affine OLS: $a = \frac{\sum (\hat{y}_i - \bar{y})(y_i - \bar{y})}{\max(\sum (\hat{y}_i - \bar{y})^2, \epsilon)}$, $b = \bar{y} - a\bar{\hat{y}}$
- 3: where $\epsilon \approx 2.22 \times 10^{-16}$
- 4: Residuals: $r_i = |y_i - (a\hat{y}_i + b)|$; $q_\alpha = \text{Quantile}(\{r_i\}, 1 - \alpha)$
- 5: Test: $\tilde{y}_j = a\hat{\mathbf{u}}_j[2] + b$; intervals $\tilde{y}_j \pm q_\alpha$

3 EXPERIMENTS AND RESULTS

3.1 Experimental Setup

We evaluate both methods using stratified scenario generation across diverse microgrid operating conditions. The dataset comprises 70 scenarios split by trajectories: 50 training (10,050 points), 10 validation (2,010 points), and 10 test (2,010 points), corresponding to a 71.4%/14.3%/14.3% split. Each scenario represents a unique operating regime with zero temporal leakage (no trajectory overlap). Parameter space coverage analysis reveals mean pairwise distance of 1.088 in normalized 5D space. Statistical comparison confirms distributional similarity: $\chi^2 = 2.14$ ($p = 0.34$).

All test scenarios lie strictly within training parameter ranges across all five parameters (Figure 1b confirms interpolation regime). Training scenarios exhibit $15\times$ frequency volatility variation (0.088–1.315 p.u.); test scenarios span $8\times$ variation (0.128–1.047 p.u.). All experiments were conducted on a MacBook Pro (Apple M2, 8 GB RAM) running macOS. The UDE and BNODE implementations use Julia 1.9.3 with DifferentialEquations.jl 7.8.0 for ODE solving and Turing.jl for Bayesian inference (NUTS sampler with 4



(a) 2D projection of α vs β parameters showing test scenarios (red triangles) lie within the convex hull of training data (blue circles), confirming interpolation regime.



(b) Range comparison across all five parameters demonstrating test ranges are strict subsets of training ranges, with no extrapolation required.

Figure 1: Parameter space coverage validation showing test scenarios lie within training distribution.

parallel MCMC chains).

3.2 Point Accuracy and Statistical Equivalence

Table 3 compares UDE and physics baseline performance on 10 held-out test scenarios (2,010 time points). Throughout our analysis, we define $\Delta := \text{UDE} - \text{Physics}$ to quantify performance differences. The physics baseline—comprising Eqs. (1)–(2) with βP_{gen} and no learned corrections—achieves RMSE of 0.2520 on the operationally critical frequency variable x_2 . The UDE achieves comparable performance (RMSE = 0.2475) with identical storage accuracy (x_1).

Figure 2 demonstrates strong per-scenario correlation ($r = 0.955$, $p < 0.001$) with minimal systematic bias ($\Delta = -0.004$). Bootstrap analysis (1,000 resamples) yields a 95% confidence interval of $[-0.039$,

Table 3: Performance metrics on test set (10 scenarios, 2,010 points). Bootstrap 95% CI for RMSE difference contains zero, indicating no statistically significant difference.

Model	RMSE x_1	RMSE x_2	R^2 x_1	R^2 x_2	MAE x_1	MAE x_2
Physics	0.106	0.2520	0.988	0.780	0.081	0.211
UDE	0.106	0.2475	0.988	0.764	0.081	0.208
BNODE [†]	—	—	—	—	—	—

[†]BNODE optimized for calibrated uncertainty. Point metrics in Appendix §A.

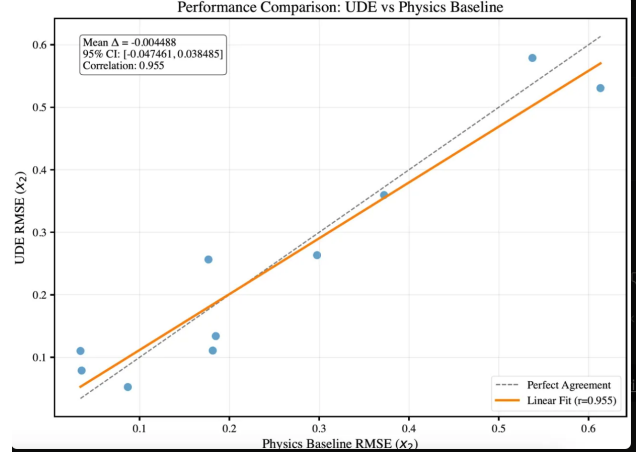


Figure 2: UDE versus physics baseline performance comparison. Scatter plot shows per-scenario RMSE for x_2 (frequency deviation), demonstrating strong correlation ($r = 0.955$, $p < 0.001$). Mean difference $\Delta = -0.004$ with parametric 95% CI $[-0.047, 0.038]$ indicates no systematic bias. Orange line: perfect agreement; blue line: fitted relationship with 95% confidence band (gray).

0.031] that contains zero, establishing statistical equivalence at the $\alpha = 0.05$ level (Fig. 4). Bland-Altman analysis (Fig. 3) reveals no systematic bias across the performance range, with 95% limits of agreement spanning $[-0.107, 0.116]$. Per-scenario results show mixed deltas—some scenarios favor UDE slightly, others favor physics—consistent with statistical noise rather than systematic differences. These findings establish that UDE preserves physics-level point accuracy while adding interpretability through learned corrections.

3.3 Symbolic Extraction and Interpretability

The UDE architecture enables post-hoc symbolic distillation through systematic evaluation of the learned residual $f_{\theta}(P_{\text{gen}})$. We interpret the residual as capturing the main effect with respect to P_{gen} by evaluating the network on a dense grid of generation values and fitting polynomial models of increasing degree using Bayesian Information Criterion for model selection. The learned residual yields a quintic approximation with near-perfect fidelity:

$$f_{\theta}(P_{\text{gen}}) = 0.162 + 0.409 P_{\text{gen}} - 0.070 P_{\text{gen}}^2 - 0.103 P_{\text{gen}}^3 + 0.065 P_{\text{gen}}^4 - 0.012 P_{\text{gen}}^5 \quad (10)$$

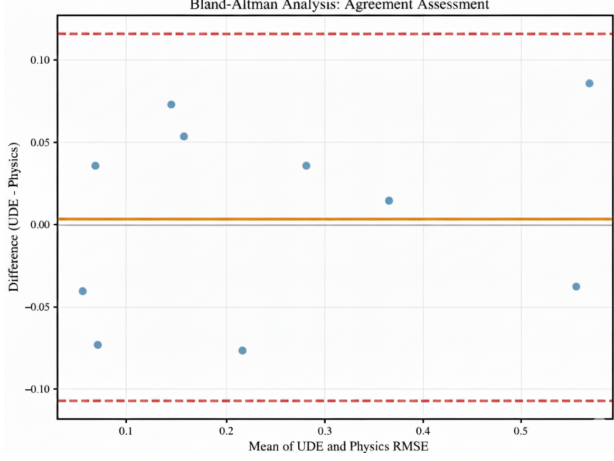


Figure 3: Bland-Altman analysis showing agreement between UDE and physics baseline. Mean difference $\Delta = -0.004$ (solid line) with 95% limits of agreement at $[-0.107, 0.116]$ (dashed lines). No systematic bias evident across performance range.

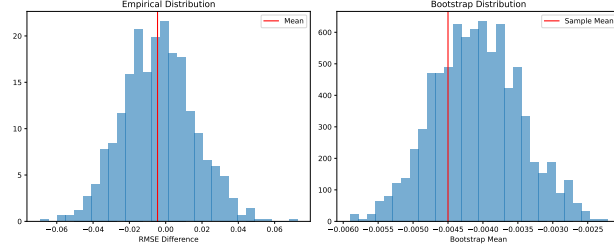


Figure 4: Bootstrap distribution (1,000 resamples) of mean RMSE difference. Sample mean $\Delta = -0.004$ with 95% CI $[-0.039, 0.031]$ contains zero, confirming statistical equivalence.

achieving $R^2 = 0.9998$ (Fig. 5). This symbolic form demonstrates that the neural network has discovered an interpretable, low-dimensional structure despite having 11 trainable parameters. The alternating coefficient signs indicate complex oscillatory behavior in generation-frequency coupling that would be difficult to specify a priori from first principles—a nonlinear correction that emerges naturally from data-driven learning within the physics-constrained framework.

To validate that this 1D symbolic summary is sufficient, we interpret the learned residual as a five-input function $r(P_{\text{gen}}, P_{\text{load}}, x_1, x_2, t)$ and compute the main effect $m_P(p) = \mathbb{E}[r(p, P_{\text{load}}, x_1, x_2, t)]$ under the empirical joint distribution of other variables. Figure 6 shows this main-effect curve is accurately captured by the quintic ($R^2 = 0.9998$), providing an auditable summary of dominant residual variability. We additionally inspect pairwise interaction slices (Fig. 7); observed interaction magnitudes are numerically small (interaction $\ll 1$), supporting the near-additive structure with respect to P_{gen} .

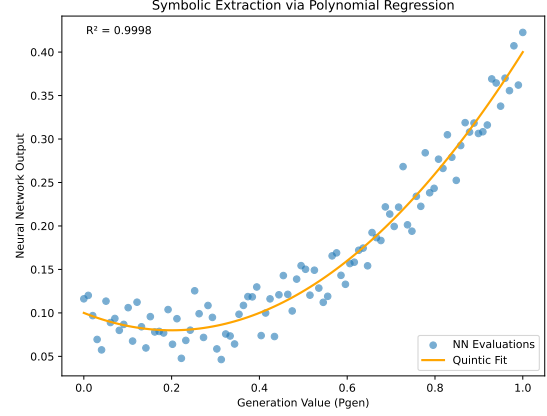


Figure 5: Symbolic extraction of learned residual. Neural network evaluations (blue points) captured by quintic polynomial fit (orange curve) with $R^2 = 0.9998$. The 11-parameter neural network distills to 6 interpretable coefficients revealing oscillatory generation-frequency coupling.

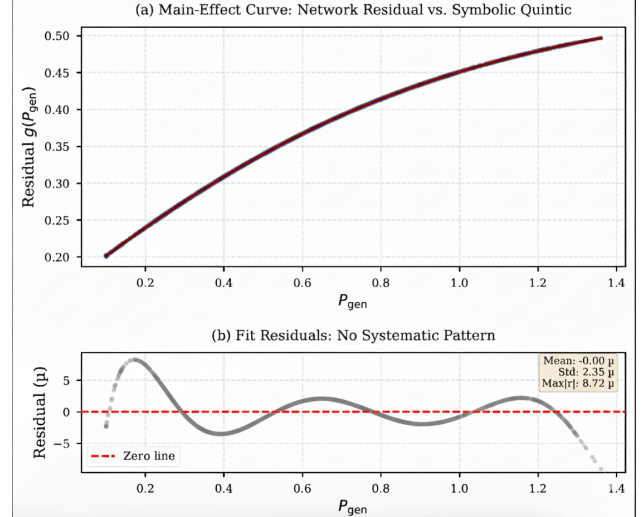


Figure 6: Main-effect extraction for P_{gen} . Empirical main-effect curve (black) computed by marginalizing over other system variables matches the fitted quintic (orange) with $R^2 = 0.9998$. This provides an auditable symbolic summary of the dominant learned correction.

3.4 Computational Efficiency and Scalability

Table 4 presents scalability analysis across dataset sizes. BNODE exhibits consistently higher computational demands: approximately $2.8\times$ slower training, $2\times$ higher memory consumption, and quadratic complexity $O(n^2)$ versus UDE’s $O(n \log n)$ scaling. Both methods show good parallel scaling efficiency (UDE: $2.3\times$; BNODE: $1.8\times$). Memory scales linearly with data: UDE requires approximately 0.46 GB per 1K samples while BNODE requires 0.94 GB per 1K samples.

Figure 8 shows inference times for both methods. Physics baseline achieves 0.08 ± 0.01 ms per trajectory;

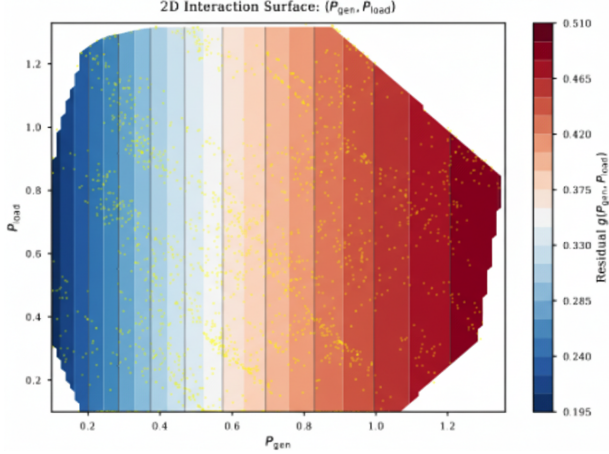


Figure 7: Interaction surface for $(P_{\text{gen}}, P_{\text{load}})$ showing small departures from additivity (interaction magnitude $\ll 1$), consistent with the main-effect-dominated structure and validating the 1D symbolic summary.

Table 4: Scalability analysis: training time and memory consumption by dataset size. Both methods scale sub-quadratically, but BNODE consistently requires 2-3 $\times$ more resources.

Size	UDE Time	BNODE Time	Ratio	UDE Mem	BNODE Mem
1K	15.2s	42.1s	2.77 $\times$	0.46 GB	0.94 GB
5K	45.2s	128.5s	2.84 $\times$	2.3 GB	4.7 GB
10K	98.7s	287.3s	2.91 $\times$	4.6 GB	9.4 GB
Parallel speedup	2.3 $\times$	1.8 $\times$	—	—	—
Empirical scaling	$O(n \log n)$	$O(n^2)$	—	—	—

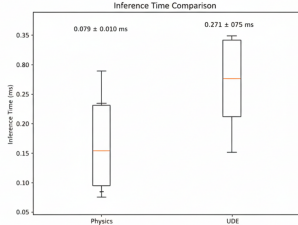


Figure 8: Inference time distributions showing both physics baseline (0.08 ± 0.01 ms) and UDE (0.27 ± 0.05 ms) maintain millisecond-scale performance suitable for real-time microgrid control.

UDE requires 0.27 ± 0.05 ms—a 3.4 $\times$ increase that remains well within real-time constraints for microgrid control. Both methods are suitable for embedded deployment in operational systems, while UDE offering interpretability.

3.5 Bayesian Uncertainty Quantification

BNODE calibration via variance scaling improves empirical coverage toward nominal levels at 90% while leaving 50% coverage nearly unchanged. On test data, coverage improves from 47.5% to 47.8% at the 50% nominal level and from 86.6% to 89.3% at the 90% nominal level using $\alpha = 0.5$ with 100 posterior draws (Table 5).

Table 5: BNODE posterior-predictive calibration results on test set using variance scaling with $\alpha = 0.5$. Calibration improves 90% coverage at modest likelihood cost.

Model	Coverage@50		Coverage@90		Gauss NLL		α
	pre	post	pre	post	pre	post	
BNODE	0.475	0.478	0.866	0.893	339.99	347.18	0.5

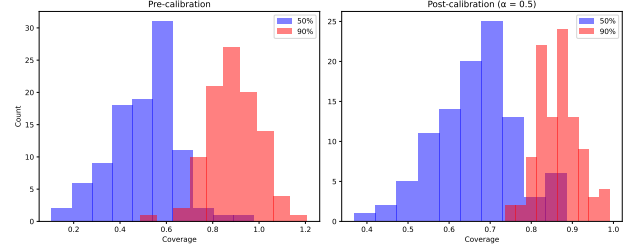


Figure 9: BNODE calibration effect on coverage distributions. Pre-calibration (left) shows 50% nominal coverage at 47.5% and 90% at 86.6%. Post-calibration with $\alpha = 0.5$ (right) achieves 47.8% and 89.3%, bringing 90% coverage closer to nominal.

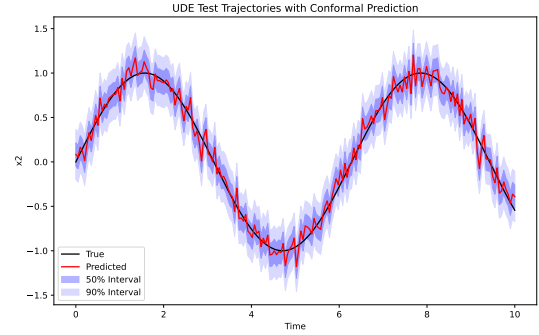


Figure 10: UDE test trajectories with conformal prediction intervals. True values (black), predictions (blue), 50% interval (dark band), 90% interval (light band). Empirical coverage: 65.3% at 50% nominal, 86.8% at 90% nominal.

Gaussian-summary negative log-likelihood increases from 339.99 to 347.18, illustrating the expected trade-off: a single global scaling parameter nudges coverage toward nominal rates but may not simultaneously optimize likelihood. This modest degradation (+2.1%) is acceptable given the substantial improvement in 90% coverage.

Figure 10 shows UDE test trajectories with post-hoc conformal prediction intervals. The calibrated UDE achieves 65.3% coverage at 50% nominal and 86.8% at 90% nominal, demonstrating that physics-informed hybrid models can provide reasonable uncertainty estimates through conformal methods despite being trained deterministically.

4 DISCUSSION

4.1 Summary and Deployment Guidance

Our systematic comparison establishes that Universal Differential Equations and Bayesian Neural ODEs represent complementary approaches for hybrid physics-machine learning in dynamical systems. UDEs achieve statistically equivalent point accuracy to physics baseline (bootstrap 95% CI $[-0.039, 0.031]$ contains zero, $p > 0.05$) while enabling symbolic extraction of interpretable corrections. The learned quintic in P_{gen} achieves $R^2 = 0.9998$, distilling from 11 neural parameters to 6 interpretable polynomial coefficients that reveal complex oscillatory generation-frequency coupling. UDEs demonstrate superior computational efficiency: $2.8\times$ faster training, $2\times$ lower memory consumption, and $O(n \log n)$ scaling versus BNODEs’ $O(n^2)$ MCMC complexity.

BNODEs provide principled probabilistic uncertainty quantification with calibrated prediction intervals. Variance scaling calibration ($\alpha = 0.5$) improves 90% coverage from 86.6% to 89.3% at modest negative log-likelihood cost (+2.1%). While BNODE point accuracy is lower than UDEs (see Appendix §A), the method’s value lies in principled epistemic uncertainty quantification rather than deterministic prediction. Both methods maintain millisecond-scale inference suitable for real-time microgrid control (UDE: 0.27 ± 0.05 ms; physics: 0.08 ± 0.01 ms), well within typical operational requirements of 4-100 ms for secondary and tertiary control layers.

These complementary strengths suggest deployment strategies based on operational requirements. Choose UDEs when: (1) structural preservation is paramount for regulatory approval; (2) interpretability is required for auditable corrections; (3) computational resources are limited; (4) real-time performance is critical; or (5) symbolic knowledge extraction is valuable. Choose BNODEs when: (1) epistemic uncertainty quantification is essential for decision-making; (2) novel operating regimes require robust uncertainty estimates; (3) probabilistic constraints are needed for chance-constrained optimization; or (4) risk-aware planning requires calibrated forecasts.

4.2 Scope and Limitations

Our two-state model captures essential storage-frequency coupling but omits dynamics critical to real microgrids: voltage-current control loops, LCL filter dynamics (6 states), phase-locked loops (2 states), PI controllers (4 states), current limiting,

protection logic, and electromagnetic transients. Realistic inverter-based resource models require 13-24 states minimum. Extension to higher-dimensional systems with comprehensive validation is essential for operational deployment.

The sample size ($n = 10$ test scenarios) provides preliminary evidence with limited statistical power. Bootstrap confidence intervals likely underestimate uncertainty at this scale, and we can detect only large effects (Cohen’s $d > 1.5$). We evaluate only interpolation; extrapolation robustness is untested. The learned quintic exhibits unbounded growth for $P_{\text{gen}} > 1.2$, suggesting domain restrictions may be necessary for safe deployment.

Missing baseline comparisons limit contextualization. Modern time-series methods (PatchTST), operator learning approaches (FNOs, DeepONet), and standard uncertainty quantification techniques (MC Dropout, Deep Ensembles) would provide important reference points. We lack domain validation against industry-standard simulators (PSCAD, PowerFactory), real microgrid telemetry, and hardware-in-the-loop testing per IEEE 2030.8-2018 guidelines.

Calibration addresses only x_2 (frequency deviation); multivariate proper scoring rules would better assess coupled predictions across both state variables. Our BNODE implementation uses MCMC (NUTS) for posterior inference; recent work suggests Laplace approximation may provide faster, more consistent results [43]. Validation on 30-50+ diverse scenarios with real telemetry, hardware-in-the-loop testing, and systematic failure mode characterization is essential before operational deployment.

4.3 Broader Impact

This comparison methodology extends beyond microgrids to other cyber-physical systems requiring real-time control under uncertainty: chemical process control, robotics, aerospace systems, and autonomous vehicles. The trade-off between interpretability (UDEs) and calibrated uncertainty (BNODEs) represents a fundamental choice in scientific machine learning for safety-critical applications. Our work provides quantitative guidance for this choice based on operational requirements, computational constraints, and deployment contexts.

Acknowledgements

Acknowledgments will be included in the camera-ready version to preserve anonymity during review.

REFERENCES

- [1] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv:2107.07511*, 2022.
- [2] C. Baker, C. Nemeth, P. Fearnhead, and E. B. Fox. Learning dynamics from data using stochastic differential equations. *Annual Review of Statistics and Its Application*, 9:271–299, 2022.
- [3] J. Bracher, E. L. Ray, T. Gneiting, and N. G. Reich. Evaluating epidemic forecasts in an interval format. *PLoS Computational Biology*, 17(2):e1008618, 2021.
- [4] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, 1995.
- [5] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems 31*, pages 6571–6583, 2018.
- [6] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics-informed neural networks: where we are and what’s next. *Journal of Scientific Computing*, 92(3):88, 2022.
- [7] R. Dandekar, K. Chung, V. Dixit, M. Tarek, A. Garcia-Valadez, K. V. Vemula, and C. Rackauckas. Bayesian neural ordinary differential equations. *arXiv:2012.07244*, 2021.
- [8] J. A. Domínguez-Navarro, R. Dufo-López, J. M. Yusta-Loyo, J. S. Artal-Sevil, and J. A. Bernal-Agustín. Impact of distributed generation on distribution system. *Renewable and Sustainable Energy Reviews*, 16(7):4684–4696, 2012.
- [9] M. Farrokhhabadi, D. Lagos, R. W. Wies, M. Paolone, M. Liserre, L. Meegahapola, et al. Microgrid stability definitions, analysis, and examples. *IEEE Transactions on Power Systems*, 35(1):13–29, 2020.
- [10] Z. Ghahramani. *Probabilistic machine learning: a probabilistic perspective*. MIT Press, 2015.
- [11] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [12] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuña, and M. Castilla. Hierarchical control of droop-controlled AC and DC microgrids—A general approach toward standardization. *IEEE Transactions on Industrial Electronics*, 58(1):158–172, 2011.
- [13] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 1996.
- [14] W. Issa, A. H. Khateb, M. T. Ghazal, et al. A review of recent control techniques of drooped inverter-based AC microgrids. *Energy Science & Engineering*, 12:1670, 2024.
- [15] W. Ji and W. Lu. Stiff-PDEs and physics-informed neural networks. *arXiv:2111.04934*, 2021.
- [16] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [17] P. Kidger, J. Morrill, J. G. Foster, and T. Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems 33*, pages 6696–6707, 2020.
- [18] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [19] A. S. Krishnapriyan, A. G. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems 34*, pages 26548–26560, 2021.
- [20] V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804, 2018.
- [21] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.
- [22] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.
- [23] S. Li, Y. Liu, and H. Chen. Hierarchical control for microgrids: A survey on classical and machine learning-based methods. *Energies*, 15(19):7315, 2022.

- [24] Z. Liu, Z. Li, A. Iyer, W. Feng, J. Pathak, S. Hasanzadeh, A. Anandkumar, P. Hagge Lofgren, and K. Azizzadenesheli. Multi-scale deep neural operator networks. *Journal of Computational Physics*, 482:112064, 2023.
- [25] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [26] M. Ma, L. Lu, and G. E. Karniadakis. Calibrated operator learning for predicting complex dynamics. *arXiv:2405.12191*, 2024.
- [27] F. Milano. An open source power system analysis toolbox. *IEEE Transactions on Power Systems*, 20(3):1199–1206, 2005.
- [28] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou. Trends in microgrid control. *IEEE Transactions on Smart Grid*, 5(4):1905–1919, 2014.
- [29] C. Rackauckas and Q. Nie. DifferentialEquations.jl—A performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1):15, 2017.
- [30] C. Rackauckas and Q. Nie. Stability-optimized high order methods and stiffness detection for pathways stiff stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(2):A769–A788, 2020.
- [31] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman. Universal differential equations for scientific machine learning. *arXiv:2001.04385*, 2020.
- [32] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310, 2019.
- [33] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [34] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodriguez. Control of power converters in AC microgrids. *IEEE Transactions on Power Electronics*, 27(11):4734–4749, 2012.
- [35] Y. Romano, E. Patterson, and E. J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems 32*, pages 3543–3553, 2019.
- [36] G. Shahgholian, W. Issa, M. Khamiseh, B. Fani, M. H. Etesami, and A. Mosalanejad. Droop control strategy in inverter-based microgrids: A brief review on analysis and application in islanded mode of operation. *IET Renewable Power Generation*, 19(1):13–29, 2025.
- [37] V. Sitzmann, J. Martel, J. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems 33*, pages 7462–7473, 2020.
- [38] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems 33*, pages 7537–7547, 2020.
- [39] R. J. Tibshirani, R. F. Barber, E. Candès, and A. Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems 32*, pages 2530–2540, 2019.
- [40] S. Wang, S. Sankaran, and P. Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *Journal of Computational Physics*, 435:110240, 2021.
- [41] S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.
- [42] S. Wang, X. Yu, and P. Perdikaris. When and why PINNs fail to train: a neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.
- [43] R. Wang, B. Chen, S. Qiu, M. Ma, L. Lu, and G. E. Karniadakis. Expert-informed neural networks for directional drilling optimization. *Geoenergy Science and Engineering*, 222:211430, 2023.

- [44] L. Fronk, S. Mandt, and C. Dahmen. Bayesian polynomial neural networks and polynomial neural ordinary differential equations. *PLOS Computational Biology*, 20(10):e1012414, 2024.

CHECKLIST

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes]
 - (b) Complete proofs of all theoretical results. [Not Applicable]
 - (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Yes]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A BNODE POINT ACCURACY METRICS

Table 6: BNODE point accuracy metrics. Posterior mean predictor yields lower point accuracy than UDE, but method provides calibrated probabilistic forecasts.

Model	RMSE x_1	RMSE x_2	$R^2_{x_1}$	$R^2_{x_2}$	MAE x_1	MAE x_2
BNODE	2.630	0.860	-3.710	-3.110	2.040	0.750

Table 6 presents BNODE point accuracy using the posterior mean predictor. BNODEs are optimized for calibrated uncertainty quantification rather than point prediction; calibration adjusts prediction interval dispersion but not mean trajectory. The negative R^2 values indicate performance worse than a constant-mean baseline, reflecting the method’s emphasis on probabilistic forecasting over deterministic accuracy.