

# Conformalized Quantum DeepONet Ensembles for Scalable Operator Learning with Distribution-Free Uncertainty

**Purav Matlia**

*Department of Computer Science  
Purdue University*

*pmatlia@purdue.edu*

**Christian Moya**

*Department of Mathematics  
Purdue University*

*cmoyacal@purdue.edu*

**Guang Lin**

*Department of Mathematics  
Department of Mechanical Engineering  
Purdue University*

*guanglin@purdue.edu*

## Abstract

Operator learning enables fast surrogate modeling of high-dimensional dynamical systems, but existing approaches face two fundamental limitations: quadratic inference complexity and unreliable uncertainty quantification in safety-critical settings. We propose Conformalized Quantum DeepONet Ensembles, a framework that addresses both challenges simultaneously. By leveraging Quantum Orthogonal Neural Networks (QOrthoNNs), we reduce operator inference complexity from  $\mathcal{O}(n^2)$  to  $\tilde{\mathcal{O}}(n)$ , enabling scalable evaluation over fine discretizations. To provide rigorous uncertainty quantification, we combine ensemble-based epistemic modeling with adaptive conformal prediction, yielding distribution-free coverage guarantees. A key challenge in ensembling is that naïve parallelism scales hardware resources linearly with the number of models. We resolve this by using Superposed Parameterized Quantum Circuits (SPQCs), which compress multiple ensemble members into a single circuit and enable simultaneous multi-model execution. Experiments on synthetic partial differential equations and real-world power system dynamics demonstrate that our approach achieves accurate predictions while maintaining calibrated uncertainty under realistic quantum noise. These results establish a practical pathway toward scalable, uncertainty-aware operator learning in quantum machine learning.

## 1 Introduction

Evaluating dynamical systems across large sets of hypothetical scenarios is central to scientific computing and safety-critical applications such as power systems. While neural operator methods such as DeepONet provide fast surrogate models for these tasks, they face two fundamental challenges: (i) scalability, due to the quadratic  $\mathcal{O}(n^2)$  cost of dense neural layers, and (ii) reliability, as standard uncertainty quantification methods either lack rigorous guarantees or incur prohibitive computational overhead. These limitations are particularly acute in settings that require both fine discretization and trustworthy predictions.

We address these challenges by combining quantum operator learning with distribution-free uncertainty quantification. Our key observation is that while quantum architectures can reduce inference complexity, uncertainty estimation via ensembles introduces a competing resource bottleneck, as naïve parallelism scales qubit requirements linearly. Resolving this tension requires a joint design of scalable inference, efficient ensembling, and calibrated uncertainty.

We make the following contributions:

- 
- **Scalable quantum operator inference.** We build on Quantum DeepONet architectures based on Quantum Orthogonal Neural Networks to achieve scalable operator learning with reduced inference complexity from  $\mathcal{O}(n^2)$  to  $\tilde{\mathcal{O}}(n)$ .
  - **Resource-efficient hybrid and superposed ensembling.** We mitigate the hardware overhead of ensembling through two architectural strategies. First, we introduce a hybrid classical-quantum architecture that reduces qubit requirements and eliminates quantum hardware noise. Second, we deploy Superposed Parameterized Quantum Circuits (SPQCs) to encode all ensemble predictions within a single circuit. Unlike sequential execution—which requires multiple distinct state preparations, data encodings, and measurements—SPQCs execute these steps exactly once. The unique architecture of the Quantum DeepONet allows these strategies to complement each other.
  - **Distribution-free uncertainty quantification.** We integrate adaptive conformal prediction with ensemble outputs to provide rigorous, finite-sample coverage guarantees.
  - **Robust evaluation under realistic noise.** We demonstrate accurate and calibrated performance across synthetic PDEs and real-world power system tasks under hardware-calibrated quantum noise models.

Unlike prior work that studies quantum operator learning or conformal prediction in isolation, our framework addresses their interaction under resource constraints. In particular, while ensembles are essential for epistemic uncertainty, their naïve parallel implementation introduces a prohibitive qubit overhead. We address this tension by jointly designing the model architecture, ensemble execution, and calibration procedure, enabling uncertainty-aware operator learning without sacrificing computational efficiency.

Traditional numerical simulations accurately analyze dynamical systems governed by high-dimensional differential equations Dahlquist & Björck (2008); Butcher (2016); Liu et al. (2019). However, repeatedly evaluating these simulators for thousands of hypothetical perturbations is computationally expensive. This cost motivates using learned surrogate operators Lu et al. (2021); Li et al. (2023); Azizzadenesheli et al. (2024); Wen et al. (2022) to approximate the solution map between infinite-dimensional function spaces directly. While these neural operator architectures provide flexibility, they introduce new challenges regarding scalability and uncertainty quantification Psaros et al. (2023); Zou et al. (2024); Azizzadenesheli et al. (2024); Xiao et al. (2025); Kerenidis et al. (2022); Landman et al. (2022).

Deep Operator Networks (DeepONet) Lu et al. (2021; 2022a,b); Lin et al. (2023) approximate solution maps by decomposing them into branch and trunk sub-networks. However, classical hidden layers within these networks rely on dense matrix-vector multiplications that incur an  $\mathcal{O}(n^2)$  computational cost Kerenidis et al. (2022); Landman et al. (2022); Goodfellow et al. (2016). This quadratic scaling is prohibitive for Monte Carlo uncertainty quantification Zhang (2021); Milanović (2017); Ye et al. (2023), which requires evaluating  $10^4$  timesteps to capture fine-scale transient dynamics. Overcoming this bottleneck requires replacing classical linear layers with a sub-quadratic inference mechanism.

Recent quantum advancements introduce scalable architectures like the Quantum Fourier Neural Operator (QFNO) Jain et al. (2024) and Quantum DeepONet Xiao et al. (2025). By replacing  $\mathcal{O}(n^2)$  classical matrix multiplications with parameterized Reconfigurable Beam Splitter (RBS) circuits operating in the unary subspace Kerenidis et al. (2022); Landman et al. (2022); Xiao et al. (2025), Quantum DeepONets reduce forward-pass complexity to  $\tilde{\mathcal{O}}(n)$ . This efficiency enables the fine discretization necessary to capture high-frequency transients. However, because quantum measurements produce probabilistic outputs requiring multiple shots, a single Quantum DeepONet yields only a noisy point prediction. Consequently, these accelerated predictions alone fail to provide the epistemic uncertainty estimates required for safety-critical deployments.

To capture true epistemic uncertainty, we rely on model diversity in ensembles and conformal prediction. Although Bayesian Neural Networks Neal (1996); Goan & Fookes (2020) or Monte Carlo dropout Gal & Ghahramani (2016) traditionally model this uncertainty, they lack the distribution-free coverage guarantees required for safety-critical tasks. Conformalized deep ensembles Angelopoulos & Bates (2023); Moya et al. (2025) resolve this by providing rigorous mathematical guarantees and model-agnostic epistemic uncertainty.

---

While methods like Quantum Conformal Prediction (QCP) Park & Simeone (2024) attempt to bypass ensemble costs by leveraging the inherent measurement shots of a single quantum circuit as a probabilistic predictor, this approach fails for quantum DeepONets as shot variance captures only hardware-level noise rather than true structural epistemic diversity. However, naïvely replicating  $L$  parallel quantum DeepONets scales qubit requirements with  $L$ . This linear scaling reintroduces a hardware bottleneck that undermines the scalability required for massive Monte Carlo evaluations. Consequently, the central challenge is preserving ensemble-based epistemic diversity without proportionally increasing resource costs.

We address these bottlenecks by introducing a unified quantum ensemble framework. This approach integrates scalable quantum operator inference, model-level ensembling, and resource-efficient execution, via hybrid classical–quantum architectures and Superposed Parameterized Quantum Circuits Patapovich et al. (2025), alongside conformal calibration to guarantee distribution-free coverage.

The rest of the paper is organized as follows. Following a review of previous work in Section 2, we formulate the problem in Section 3, Section 4 details the quantum circuit design and ensembling methodology, Section 5 establishes the conformal prediction framework, while Section 6 evaluates the framework’s predictive accuracy and uncertainty calibration. Finally, we discuss limitations in Section 7. For clarity, a summary of nomenclature is provided in Appendix B, which readers may consult throughout.

## 2 Previous Work

**Operator learning and scientific machine learning.** The scientific machine learning community has introduced various data-driven frameworks for approximating the dynamic response of systems governed by ordinary and partial differential equations Karniadakis et al. (2021); Baker et al. (2019). Early efforts focused on data-driven surrogates that either learn discrete-time evolution maps or infer continuous-time governing equations Qin et al. (2019); Raissi et al. (2018), and on sparse-regression schemes that recover governing equations directly from data Brunton et al. (2016). To leverage physical priors and reduce data requirements, Physics-Informed Neural Networks (PINNs) Raissi et al. (2019) encode the governing equations as soft constraints in the loss, with extensions to stiff Ji et al. (2020), multi-scale Leung et al. (2022), and differential–algebraic systems, including applications in power grids Moya & Lin (2023); Huang & Wang (2023) and biology Yazdani et al. (2020). Because PINNs typically require retraining whenever problem parameters change, attention has shifted to operator learning, in which neural architectures approximate mappings between infinite-dimensional function spaces. Deep Operator Networks (DeepONet) Lu et al. (2021; 2022a;b); Lin et al. (2023) and Fourier Neural Operators Li et al. (2023); Wen et al. (2022); Azizzadenesheli et al. (2024) have demonstrated strong generalization across parametric PDE families and complex multi-physics applications, motivating their use as surrogates for safety-critical simulations such as power-system transient analysis Moya et al. (2023).

**Quantum machine learning for scientific computing.** Parallel to these classical advances, quantum machine learning has progressed toward learning solutions of ordinary and partial differential equations, primarily through variational and hybrid quantum–classical formulations. Variational quantum PDE solvers Joo & Moon (2021) and approaches such as quantum lattice-Boltzmann methods augmented with multi-fidelity neural correction Jacob et al. (2025) extend these ideas to both linear and nonlinear regimes. Quantum Orthogonal Neural Networks (QOrthoNN) Kerenidis et al. (2022); Landman et al. (2022) replace dense classical layers with parameterized quantum circuits that implement orthogonal transformations, yielding improved asymptotic scaling, and have been empirically tested on medical image classification tasks Landman et al. (2022). Building on the QOrthoNN, architectures like Quantum DeepONet Xiao et al. (2025) extend operator learning to quantum settings, while independent approaches such as Quantum Fourier Neural Operators (QFNO) Jain et al. (2024) have been evaluated on a range of benchmark PDE families. Quantum scientific machine learning has also replicated paradigms such as physics-informed neural networks Trahan et al. (2024); Farea et al. (2025).

**Uncertainty quantification for scientific machine learning.** Reliable deployment of neural surrogates in safety-critical settings demands rigorous uncertainty quantification Psaros et al. (2023); Zou et al. (2024). Classical Bayesian neural networks MacKay (1992); Neal (1996) model distributions over parameters

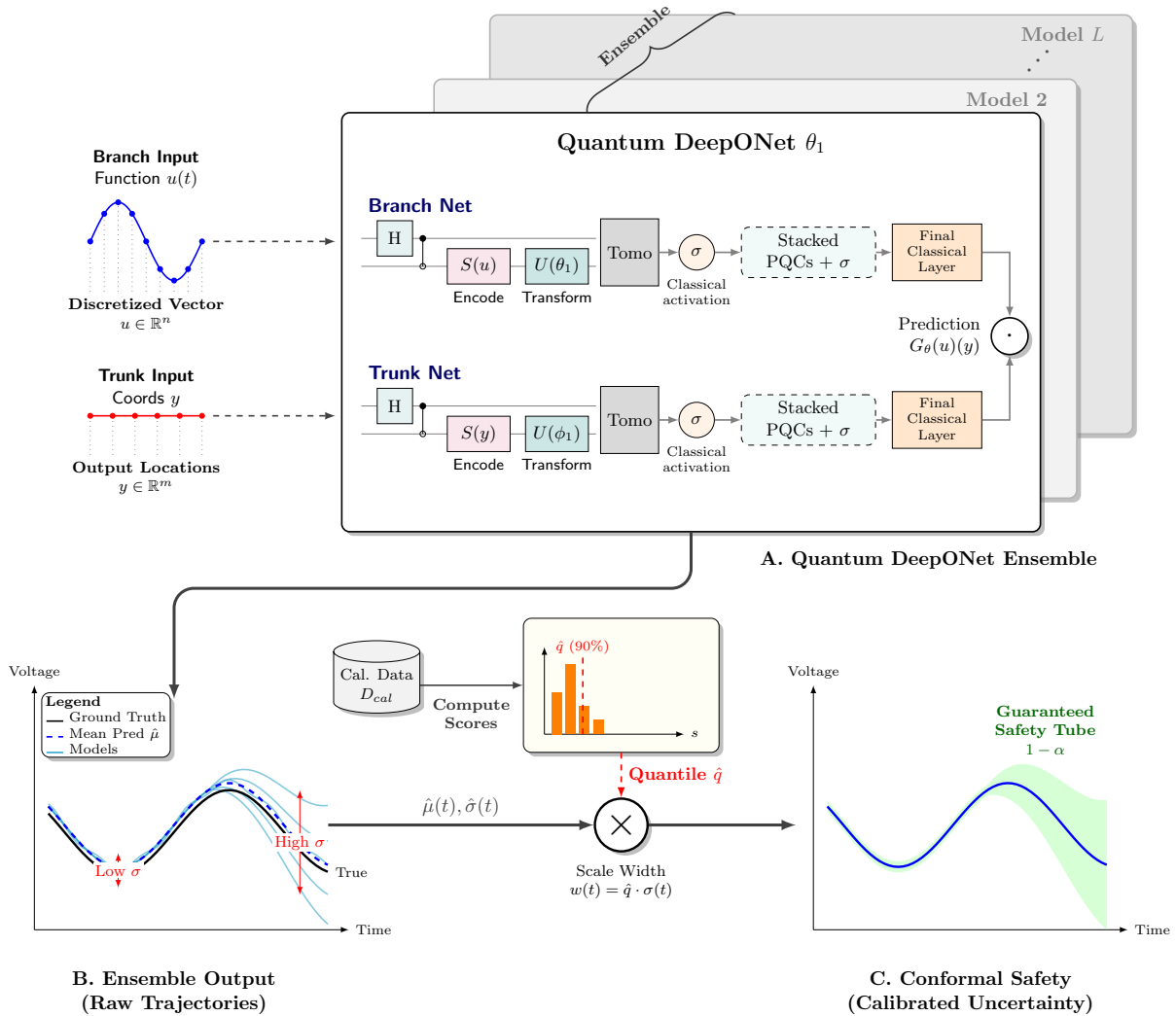


Figure 1: Overview of the conformalized quantum DeepONet ensemble framework. (A) quantum DeepONet ensemble: an ensemble of operators decomposed into two subnetworks implemented using quantum orthogonal neural networks. (B) ensemble output: predictions are aggregated to compute mean prediction  $\hat{\mu}(t)$  and standard deviation  $\hat{\sigma}(t)$  for uncertainty. (C) conformal safety: using calibration data  $\mathcal{D}_{cal}$ , nonconformity scores define threshold  $\hat{q}$  producing safety tube  $C_\alpha(\mathbf{u}, t)$  with coverage  $1 - \alpha$ .

but do not scale well or provide mathematical guarantees. Alternative approaches include deep ensembles Lakshminarayanan et al. (2017); Psaros et al. (2023), Bayesian PINNs Yang et al. (2021), and probabilistic CNNs for PDEs Winovich et al. (2019). For a comprehensive review of these deep learning UQ techniques, we refer readers to Abdar et al. (2021) and Psaros et al. (2023). These methods generally lack distribution-free coverage guarantees, motivating the adoption of conformal prediction Vovk et al. (1999; 2022); Angelopoulos & Bates (2023) to provide rigorous intervals for neural operators. Recent work further extends conformal methodology to the operator learning paradigm Moya et al. (2025). In the quantum setting, Park & Simeone (2024) applied conformal prediction to quantum machine learning, but uncertainty-aware quantum operator learning remains largely unexplored.

### 3 Problem Formulation

We aim to learn an operator  $\mathcal{G}$  that maps input functions to pointwise predictions. We train this operator using a supervised dataset  $\mathcal{D} = \{(\mathbf{u}_i, \mathbf{y}_{ij}, s_{ij})\}_{i=1, j=1}^{N, M_i}$ , where  $\mathbf{u}_i = (u(x_1), \dots, u(x_{d_u}))^\top \in \mathbb{R}^{d_u}$  represents a discretized input function evaluated at  $d_u$  fixed sensor locations. The variable  $\mathbf{y}_{ij} \in \mathbb{R}^{d_y}$  denotes a spatial or temporal query coordinate, and  $s_{ij} = \mathcal{G}(\mathbf{u}_i)(\mathbf{y}_{ij}) \in \mathbb{R}$  represents the corresponding ground-truth value of the operator evaluated at that specific coordinate. To solve this operator learning problem, we use the Deep Operator Network (DeepONet) Lu et al. (2021; 2022a;b); Lin et al. (2023) as our foundational approximation architecture.

The DeepONet architecture decomposes the target operator into two sub-networks: a branch network that encodes discretized input functions and a trunk network that embeds spatial or temporal query coordinates. For any generic input function  $\mathbf{u}$  and query coordinate  $\mathbf{y}$ , the network approximates the target operator by computing the inner product of the  $p$ -dimensional branch output vector,  $\mathbf{b}$ , and trunk output vector,  $\mathbf{t}$ :

$$\mathcal{G}(\mathbf{u})(\mathbf{y}) \approx \sum_{k=1}^p b_k t_k = \mathbf{b}^\top \mathbf{t} \in \mathbb{R}$$

We optimize the combined trainable parameters,  $\theta \in \mathbb{R}^{d_\theta}$ , of both sub-networks simultaneously by minimizing the Mean Squared Error (MSE) across the dataset:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \frac{1}{M_i} \sum_{j=1}^{M_i} (\mathcal{G}_\theta(\mathbf{u}_i)(\mathbf{y}_{ij}) - s_{ij})^2$$

where  $d_\theta$  is the total number of parameters, and the point-wise predictions and ground truths  $s_{ij}$  reside in  $\mathbb{R}$ . Although this architecture is highly expressive, it introduces a computational bottleneck. Assuming an input size of  $d_{in}$  and a uniform hidden layer width of  $n$ , the initial layer maps the input dimension to  $n$  at a cost of  $\mathcal{O}(d_{in} \cdot n)$ . Subsequent dense hidden layers rely on  $\mathcal{O}(n^2)$ . At test time, when the learned operator must be evaluated across thousands of hypothetical conditions in a scenario batch, this quadratic scaling with respect to the network width bounds the total computational budget.

Specifically, offline batch analysis requires evaluating the learned operator across  $N$  distinct hypothetical scenarios. To establish an upper bound on computational requirements, let  $M = \max_i(M_i)$  denote the maximum number of spatial or temporal query coordinates evaluated per scenario. By caching the branch network’s representation, we evaluate it only once per scenario, whereas the trunk network must be evaluated  $M$  times. This yields a per-scenario cost of  $\mathcal{O}(n^2 + M \cdot n^2)$  where  $n$  is the size of the hidden layers. Consequently, the total computational cost across the entire batch scales as  $\mathcal{O}(N \cdot M \cdot n^2)$ . This explosion in complexity is strictly prohibitive for fine discretizations. Furthermore, any uncertainty quantification method that requires multiple forward evaluations, such as ensembling or Monte Carlo sampling, will linearly multiply this massive baseline cost. Therefore, achieving robust uncertainty quantification at scale requires redesigning the inference mechanism to bypass this classical quadratic bottleneck.

To guarantee reliable uncertainty bounds without exacerbating the quadratic deployment cost, we formalize our objective using conformal prediction. Specifically, for a given input function  $\mathbf{u}$  and query coordinate  $\mathbf{y}$ , we seek to generate adaptive prediction sets,  $C_\alpha(\mathbf{u}, \mathbf{y})$ , that achieve a strict, user-specified miscoverage rate,  $\alpha$ . Generating these rigorous, distribution-free bounds while remaining computationally feasible for massive scenario batches requires overcoming the classical scaling limits. Simultaneously achieving sub-quadratic operator inference and mathematically calibrated uncertainty defines the core design constraints of our proposed methodology.

## 4 Method

### 4.1 Design principles

**Complexity** To eliminate the classical  $\mathcal{O}(n^2)$  inference bottleneck, Quantum Orthogonal Neural Networks (QOrthoNNs) Kerenidis et al. (2022); Landman et al. (2022) replace dense layers with three sub-circuits: a

data loader  $S(\cdot)$ , a parameterized unitary  $U(\cdot)$ , and tomography. Tomography is required to extract the final classical output from the evolved quantum state. Emulating a classical dense layer that executes an  $m \times n$  linear transformation using a QOrthoNN requires  $1 + \max(m, n)$  qubits; to establish a direct equivalence with classical hidden layers of width  $n$ , we evaluate the  $n \times n$  case utilizing  $n + 1$  qubits. The underlying circuit is built using Reconfigurable Beam Splitter (RBS) gates Kerenidis et al. (2022); Landman et al. (2022); Xiao et al. (2025), which require only nearest-neighbor connectivity and admit an  $\mathcal{O}(1)$  decomposition into standard hardware basis gates.

The sequential forward pass begins with the data loader  $S(\cdot)$ , which, when parameterizing an  $n$ -dimensional input vector, requires a circuit depth of  $n - 1$ . Subsequently, the pyramidal unitary  $U(\cdot)$  executing an  $m \times n$  transformation is bounded by a maximum depth of  $2 \max(m, n) - 3$ ; for our specific  $n \times n$  configuration, this simplifies to  $2n - 3$ . Finally, the tomography sub-circuit requires two additional  $S(\cdot)$  data loaders alongside an  $\mathcal{O}(1)$  supplementary gates (X, CNOT, and Hadamard). Cumulatively, this bounds the total circuit depth to  $\mathcal{O}(n)$ , which can be further compacted by pipelining independent operations. To extract the classical output, the system requires  $\mathcal{O}(\log(n)/\delta^2)$  quantum measurements to bound the finite-sampling error below a threshold  $\delta$ . Consequently, the total forward-pass time complexity drops to  $\tilde{\mathcal{O}}(n/\delta^2)$ , resolving the classical scalability bottleneck.

**Uncertainty Quantification** Although recent work explores leveraging the stochasticity of quantum measurements to quantify uncertainty Park & Simeone (2024), this methodology is flawed in our case, where epistemic uncertainty cannot be inferred from shot noise. As demonstrated by Xiao et al. (2025), the standard deviation for a single output component  $o_j$  of a QOrthoNN scales as:

$$\sigma[o_j] \propto \frac{1}{\sqrt{N_{\text{shots}}}}$$

Because this variance is strictly proportional to  $\frac{1}{\sqrt{N_{\text{shots}}}}$ , it vanishes as the measurement budget increases, regardless of input difficulty or out-of-distribution shifts. To capture genuine epistemic uncertainty arising from sparse training data, we must instead construct ensembles of independently trained Quantum DeepONets. However, naïvely executing  $L$  independent quantum models scales qubit requirements linearly, introducing a prohibitive hardware bottleneck.

**Ensembling** Naïvely executing  $L$  quantum operators in parallel demands  $\mathcal{O}(L \cdot n)$  qubits. To mitigate this overhead, we apply two complementary resource-reduction strategies. First, a hybrid classical-quantum architecture replaces the low-evaluation-frequency sub-network—such as a branch network evaluated only once per trajectory—with a classical counterpart. For instance, replacing a branch network that is evaluated only once per trajectory with a classical dense network yields a total computational cost of  $\mathcal{O}(N \cdot n^2 + N \cdot M \cdot n \log n)$  where  $N$  is the number of input trajectories and  $M$  is the number of query coordinates per trajectory. Given  $M \gg n$ , the  $\mathcal{O}(n^2)$  classical penalty is safely amortized across the batch. This targeted substitution eliminates the dominant quantum noise contribution of the low-evaluation-frequency sub-network entirely, while rigorously preserving the  $\tilde{\mathcal{O}}(n)$  quantum speedup for the high-frequency trunk network evaluations. Second, to efficiently scale the ensemble itself, we deploy Superposed Parameterized Quantum Circuits (SPQCs) Patapovich et al. (2025). By encoding all  $L$  ensemble members into a unified circuit via uniformly controlled rotations conditioned on address qubits, SPQCs evaluate the entire ensemble simultaneously. This architecture requires only  $n + 1 + \lceil \log_2(L) \rceil$  qubits per sub-network at the cost linear  $\mathcal{O}(L \cdot n)$  circuit depth scaling Patapovich et al. (2025). Guided by these design constraints, we now detail the specific quantum architecture, training procedure, and ensembling methodologies.

## 4.2 Quantum DeepONet Architecture

Each Quantum DeepONet consists of two sub-networks: a branch Quantum Orthogonal Neural Network (QOrthoNN) that processes the input function  $\mathbf{u}$ , and a trunk QOrthoNN that processes the query coordinate  $\mathbf{y}$ . Within these sub-networks, Parameterized Quantum Circuits (PQCs) execute orthogonal linear transformations, which are subsequently passed through classical non-linear activations. To remove the strict orthogonality restrictions imposed by unitary quantum operations and increase the model’s expressive capacity, a final unconstrained classical linear layer caps each QOrthoNN. The fundamental hardware building

block enabling these quantum transformations is the Reconfigurable Beam Splitter (RBS) gate, defined by the unitary matrix:

$$U_{\text{RBS}}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

When applied to a general two-qubit state  $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ , the RBS gate modifies the amplitudes of only the unary  $|01\rangle$  and  $|10\rangle$  components, mapping the state to  $\alpha|00\rangle + (\beta \cos \theta + \gamma \sin \theta)|01\rangle + (-\beta \sin \theta + \gamma \cos \theta)|10\rangle + \delta|11\rangle$ . Because it never maps states with zero or two ones into states with a single one, this preservation mechanism allows the entire QOrthoNN to operate strictly within the unary subspace. Consequently, the tomography procedure for extracting the classical output assumes a superposition of unary states, which supports error mitigation techniques.

To clarify the mechanics of a single QOrthoNN hidden layer, we briefly detail the evolution of the quantum state for an  $n \times n$  transformation, deferring the generalized  $m \times n$  mathematical formulation to Appendix A. To encode classical data onto the quantum circuit, QOrthoNNs use unary amplitude encoding. Because quantum amplitude encoding requires the input vector to have a strict unity  $\ell_2$ -norm, raw classical data must be pre-processed. If  $\mathbf{x}$  is the initial input vector, it is min-max normalized into  $[-1, 1]$ , scaled by  $1/\sqrt{n}$ , and appended with a slack variable,  $\sqrt{1 - \sum_{i=1}^{n-1} (x_i/\sqrt{n})^2}$ , to preserve information about its original magnitude while enforcing the unity norm constraint. For subsequent hidden layers, intermediate outputs are simply scaled by  $1/\sqrt{n}$ .

Once the  $n$ -dimensional normalized input vector  $\mathbf{x} \in \mathbb{R}^n$  is prepared, the PQC executes the orthogonal linear transformation  $\mathbf{W} \in \mathbb{R}^{n \times n}$  using  $n$  data qubits and one ancillary qubit. First, a data-dependent unitary  $S(\mathbf{x})$  loads the input vector onto the data register via amplitude encoding, yielding the quantum state:

$$|\psi\rangle_{\text{data}} = x_1|10 \cdots 0\rangle + x_2|01 \cdots 0\rangle + \cdots + x_n|00 \cdots 1\rangle = \sum_{i=1}^n x_n |\mathbf{e}_i\rangle$$

where  $\mathbf{e}_i$  denotes a unary computational basis state. This is followed by a pyramidal parameterized unitary  $U(\boldsymbol{\theta})$  that emulates the weight matrix  $\mathbf{W}$ , evolving the data register to:

$$|\psi\rangle = \sum_{j=1}^n \sum_{i=1}^n W_{ji} x_i |\mathbf{e}_j\rangle$$

This operation effectively performs a parallel matrix-vector multiplication directly on the quantum amplitudes. Finally, tomography recovers these transformed amplitudes back into classical floating-point values. To achieve this, the tomography sub-circuit utilizes the ancilla qubit alongside two additional  $S$  data loaders. The details of the tomography sub-circuit and quantum state evolution are outlined in Appendix A. This extracted classical output is then passed through standard non-linear activations to complete the hidden layer sequence. A QOrthoNN is a stack of these “quantum layers” followed by a final classical layer, effectively emulating a standard neural network.

Having established the forward-pass mechanics, we must address optimization. To ensure stable convergence and avoid hardware decoherence, we bypass training directly on Noisy Intermediate-Scale Quantum (NISQ) hardware Bharti et al. (2022); Preskill (2018). The feasibility of classical training is enabled from the fact that these quantum operations mathematically equate to an exact classical orthogonal linear transformation,  $\mathbf{W}$ . Because this transformation is intrinsically parameterized by the Reconfigurable Beam Splitter (RBS) rotation angles,  $\theta$ , we can natively emulate the network’s exact mathematical behaviour within standard deep learning frameworks, circumventing the need for expensive quantum state evolution simulations. During training, standard automatic differentiation directly optimizes the RBS rotation angles instead of the weight matrix elements in  $\mathcal{O}(n^2)$  time Kerenidis et al. (2022); Landman et al. (2022). By navigating the optimization trajectory strictly through this angular parameter space, the network guarantees strict mathematical orthogonality by construction. This circumvents the computationally prohibitive  $\mathcal{O}(n^3)$  Singular

Value Decomposition (SVD) or Stiefel manifold projection steps required to maintain orthogonality in traditional classical networks Kerenidis et al. (2022); Landman et al. (2022); Jia et al. (2019). Once the angles are optimized classically, these learned parameters are deployed to the PQC to execute the sub-quadratic inference.

This strict mathematical orthogonality provides two critical optimization advantages. First, by preserving eigenvalue magnitudes across deep transformations, the orthogonal constraint prevents the vanishing and exploding gradients that can destabilize deep architectures, thereby promoting robust generalization Kerenidis et al. (2022); Landman et al. (2022). Second, within the quantum domain, this orthogonality reshapes the loss landscape. By restricting the optimization trajectory with the orthogonality constraint rather than exploring the exponentially large full Hilbert space, the architecture systematically circumvents the barren plateaus of general PQCs Wang et al. (2021); McClean et al. (2018); Schumann et al. (2024).

### 4.3 Ensemble Construction

To induce predictive diversity, we train each ensemble member,  $\mathcal{G}_{\theta_m}$ , independently on the full dataset using distinct random shuffling and parameter initializations. At inference, the resulting ensemble mean,  $\mu(\mathbf{u})(\mathbf{y}) = \frac{1}{L} \sum_{m=1}^L \mathcal{G}_{\theta_m}(\mathbf{u})(\mathbf{y})$ , provides the central prediction, while the ensemble standard deviation,  $\sigma(\mathbf{u})(\mathbf{y})$ , quantifies the epistemic disagreement. Although we evaluated non-parametric bootstrapping (bagging) as an alternative mechanism, we observed no performance gain. Consistent with classical deep learning theory Lakshminarayanan et al. (2017); Lee et al. (2015), because these high-capacity models require maximal training data, the loss of unique samples inherent to bootstrapping outweighs any diversity benefits. Finally, because sequentially executing this identically distributed ensemble remains prohibitive on near-term hardware, we address resource efficiency through two specialized execution strategies.

### 4.4 Hybrid Classical–Quantum Architectures

The DeepONet’s inherent branch-trunk decomposition allows us to selectively replace one sub-network with a classical counterpart to optimize resource efficiency. Specifically, we replace the low-evaluation-frequency component—typically the branch network, which is evaluated only once per input trajectory—to eliminate its quantum hardware noise penalty. This classical substitution incurs a one-time  $\mathcal{O}(n^2)$  computational penalty per trajectory. However, because the total evaluation cost scales as  $\mathcal{O}(N \cdot n^2 + N \cdot M \cdot n \log n)$ , this substitution remains strictly optimal whenever the number of query coordinates per trajectory significantly exceeds the network width ( $M \gg n$ ). Under this condition, the  $\mathcal{O}(n^2)$  classical penalty is safely amortized, successfully isolating and eliminating the dominant hardware noise contribution while rigorously preserving the  $\tilde{\mathcal{O}}(n)$  quantum acceleration for the high-frequency trunk network.

Alternatively, one could reduce resource demands by sharing a single sub-network across all ensemble members. However, this strategy sacrifices predictive diversity. By forcing the independent, unshared sub-networks to adapt to a fixed intermediate output, this shared representation acts as an informational bottleneck. Thus, it artificially constrains the ensemble’s variance and degrades uncertainty quantification by yielding overconfident predictions.

### 4.5 Superposed Parameterized Quantum Circuits

To eliminate the linear qubit scaling associated with parallel execution, our second strategy employs Superposed Parameterized Quantum Circuits (SPQCs) Patapovich et al. (2025). By efficiently encoding all  $L$  ensemble members into a single unified circuit, SPQCs require only  $n + 1 + \lceil \log_2(L) \rceil$  qubits per sub-network for  $n \times n$  transformations at the cost linear  $\mathcal{O}(L \cdot n)$  circuit depth scaling. We realize this architecture through a cascade of uniformly controlled rotations conditioned on address qubits initialized in a uniform superposition:

$$|\psi\rangle = \frac{1}{\sqrt{L}} \sum_{j=1}^L |j_{addr}\rangle \otimes U(\boldsymbol{\theta}^{(j)})S(\mathbf{x}^{(j)})|0_D\rangle^{\otimes n}.$$

By conditioning the data loading unitaries,  $S(\mathbf{x}^{(j)})$ , on these address bits, we ensure each specific model,  $U(\boldsymbol{\theta}^{(j)})$ , processes its correct input data. Correspondingly, a modified tomography protocol extracts the independent outputs of all  $L$  models simultaneously. This multiplexing enables the system to perform state preparation, unitary evolution, and measurement exactly once for the entire ensemble. Although the requisite controlled operations inevitably increase circuit depth and exacerbate accumulated quantum noise on near-term devices, SPQCs provide a highly scalable, resource-efficient execution strategy as physical hardware coherence improves.

While executing models sequentially on a single circuit demands  $L$  distinct state preparations and measurement cycles—scaling the total runtime overhead linearly with the ensemble size—the SPQC executes these costly input/output operations exactly once. By circumventing these repeated preparation and readout bottlenecks, this multiplexed approach yields significant runtime advantages over sequential execution.

## 5 Uncertainty Quantification

To translate the raw ensemble dispersion into mathematically calibrated bounds, we apply adaptive conformal prediction Angelopoulos & Bates (2023); Vovk et al. (1999; 2022); Shafer & Vovk (2008); Papadopoulos et al. (2002). During the calibration phase, we compute an adaptive nonconformity score,  $r_{ij}$ , for each data point  $(\mathbf{u}_i, \mathbf{y}_{ij}, s_{ij})$  by normalizing the absolute prediction error by the local ensemble standard deviation:

$$r_{ij} = \frac{|s_{ij} - \mu(\mathbf{u}_i)(\mathbf{y}_{ij})|}{\sigma(\mathbf{u}_i)(\mathbf{y}_{ij}) + \epsilon}$$

By calculating the appropriate empirical quantile,  $\hat{q}$ , of these nonconformity scores, we construct an adaptive uncertainty bound around the predicted ensemble mean. For a given input function  $\mathbf{u}$  and a query coordinate  $\mathbf{y}$ , any candidate true output value  $v \in \mathbb{R}$ , the prediction set is defined as:

$$C_\alpha(\mathbf{u}, \mathbf{y}) = \{v \in \mathbb{R} : |v - \mu(\mathbf{u})(\mathbf{y})| \leq \hat{q} \cdot \sigma(\mathbf{u})(\mathbf{y})\}$$

Crucially, when the calibration and test examples remain exchangeable, this framework provides a distribution-free mathematical guarantee of finite-sample pointwise marginal coverage Angelopoulos & Bates (2023); Shafer & Vovk (2008). This ensures that at any specific query coordinate, the true output value falls within the prediction set with a strict, user-specified probability:

$$P(s_{ij} \in C_\alpha(\mathbf{u}_i, \mathbf{y}_{ij})) \geq 1 - \alpha$$

The theoretical validity of these distribution-free coverage bounds strictly requires the calibration and test examples to be exchangeable Angelopoulos & Bates (2023); Shafer & Vovk (2008). However, real-world deployments introduce challenges to this assumption, which we explicitly acknowledge. First, physical quantum hardware noise—such as fluctuating depolarizing channels, readout errors, and thermal relaxation—can drift between the calibration and test phases, inherently threatening exchangeability. While we currently evaluate conformal prediction assuming these noise parameters remain static between calibration and testing, future deployments on temporally drifting hardware will formally violate this premise. Second, evaluating sequential tasks with temporal dependencies, such as online prediction with sliding windows, violates this statistical assumption by definition. Rather than restricting our analysis to idealized conditions, we treat the non-exchangeable online setting as a rigorous stress test for our uncertainty calibration. We evaluate the framework’s empirical coverage under both exchangeable batch settings (with static noise) and this non-exchangeable sequential environment, deferring the derivation of formal theoretical bounds for the latter to future work.

## 6 Experiments

### 6.1 Setup and Evaluation Metrics

We evaluate the predictive accuracy of our framework using relative  $L_2$  error, and we assess the quality of uncertainty calibration through empirical coverage, average interval width, and peak uncertainty across both

Table 1: Summary of model performance across synthetic experiments under ideal simulation with 90% target coverage. The metrics from left to right are Relative  $L_2$  Error (%), Coverage (%), Average Width, and Peak Uncertainty.

Experiment	Rel. $L_2$ Err. (%)	Cov. (%)	Avg. Width	Peak Unc.
Antiderivative (Ens. Size = 4)	0.46	88.40	0.004	0.044
Antiderivative (Ens. Size = 8)	0.46	92.13	0.005	0.080
Advection (Ens. Size = 4)	2.38	89.36	0.062	0.751
Advection (Ens. Size = 8)	2.28	88.99	0.053	0.621

synthetic and real-world operator-learning tasks. Establishing an ideal baseline for these metrics on large quantum circuits, those exceeding 20 qubits, presents a severe computational bottleneck, as traditional state-vector simulations incur exponential memory costs. To bypass this limitation, we execute ideal simulations by directly evaluating the mathematically equivalent classical Orthogonal Neural Network, perfectly mimicking the noiseless quantum state evolution while avoiding exponential overhead.

Beyond these ideal baselines, we evaluate the framework under varying degrees of quantum noise. To conduct hyperparameter ablations efficiently, we first use a simplified gate-based noise model incorporating finite-sampling error and depolarizing channels. For single- and two-qubit gates, the depolarizing error on a quantum state  $\rho$  is modeled as

$$\mathcal{E}(\rho) = (1 - \lambda)\rho + \lambda \text{Tr}[\rho] \frac{I}{2^n},$$

with noise strength  $\lambda$  scaled by 0.8 for two-qubit operations to equalize error rates across 1 and 2-qubit depolarizing noise channels. We evaluate circuits under this model by transpiling them to IBM’s Eagle basis gate set {ECR,  $R_z$ , SX, X, I} AbuGhanem (2025); Xiao et al. (2025), assuming nearest-neighbour connectivity. To accelerate simulations, we perform multi-nomial sampling of the evolved density matrix, which perfectly correlates with computationally expensive per-shot simulations as outlined in Appendix D. We rigorously validate near-term hardware readiness using a Qiskit Aer noise model Javadi-Abhari et al. (2024). This realistic model simulates single-qubit readout errors, depolarizing errors, and thermal relaxation. For this validation, circuits are strictly transpiled according to the reported calibration parameters, basis gates, and topological coupling maps of three representative quantum processing units: IBM Brisbane (127-qubit Eagle r3), IBM Torino (133-qubit Heron r1), and IBM Marrakesh (156-qubit Heron r2) IBM Quantum (2026). The snapshots of calibration data used in our experiments can be found in E.

Operating in these realistic, noisy environments needs robust error mitigation. Because the QOrthoNN architecture structurally encodes data exclusively within the unary subspace, the valid quantum state must always maintain a Hamming weight of exactly one. We exploit this strict physical constraint to mitigate hardware errors through post-selection. During tomography, we simply discard any measurement shot that collapses into an invalid, non-unary state. By trading a larger total shot budget for higher fidelity readouts, this post-selection systematically filters out corruption, reducing the impact of hardware noise on the final operator predictions.

## 6.2 Synthetic Benchmarks

### 6.2.1 Antiderivative Operator

We first evaluate our framework on a synthetic antiderivative operator,  $\mathcal{F} : v \mapsto u$ . This operator maps an input function  $v(x)$  to its integral  $u(x)$  over the domain  $x \in [0, 1]$ , defined by the differential relation  $\frac{du(x)}{dx} = v(x)$  subject to the initial condition  $u(0) = 0$ . To generate the dataset, we sample smooth input functions from a Gaussian Random Field (GRF) prior,  $v \sim \mathcal{GP}(0, k_l(x_1, x_2))$ . The covariance is governed by a squared exponential kernel,  $k_l(x_i, x_j) = \exp(-\frac{d(x_i, x_j)^2}{2l^2})$ , where the length scale  $l$  controls the smoothness of the sampled functions. For this task, we deploy an ensemble of  $L = 8$  independent models, using a 2-layer,

Table 2: Architecture and hardware specifications for the quantum DeepONet ensembles for the synthetic benchmarks. Width denotes the number of neurons per layer. Params and circuit depth report the maximum per-layer values for trainable RBS angles ( $\theta$ ) and transpiled PQC depth (on IBM Heron/Eagle), respectively. All models use the SiLU activation function. (Config: R = ResNet, F = Fourier features). The number of qubits used for a single model is  $d_u + 2$ .

Experiment	$d_u$	$d_y$	Layers	Width	Params	Heron	Eagle	Config	Ens. Size
Antiderivative	10	1	2	10	55	165	321	–	4/8
Advection	20	1	7	20	210	305	601	R	4/8

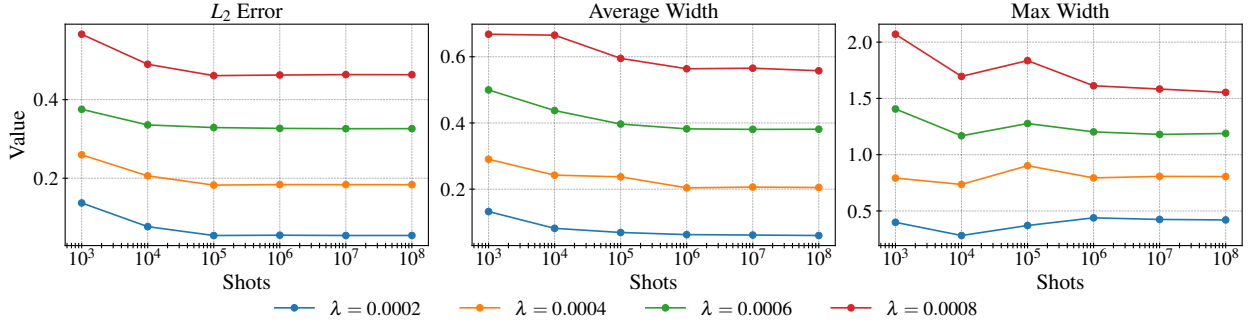


Figure 2: Impact of depolarizing noise  $\lambda$  (represented by different colours) and finite-sampling error (x-axis) on Antiderivative prediction performance. All three metrics:  $L_2$  Error (left), Average Width (middle), and Max Width (right) improve with increasing shot counts and yield better performance at lower noise levels.

10-neuron QOrthoNN architecture. As detailed in Table 2, this model requires only 55 trainable parameters per layer and transpiles to a maximum circuit depth of 165 on the IBM Heron topology and 321 on IBM Eagle. Under ideal noiseless simulation, this quantum ensemble successfully learns the operator, achieving a 0.46% relative  $L_2$  error and 92.13% empirical coverage for a target rate of  $1 - \alpha = 0.90$ .

We further studied the effects of depolarizing noise on the approximation of the antiderivative operator with  $L = 8$  ensemble members each composed of the same QOrthoNN architecture described above. We observed that accuracy and uncertainty metrics follow a general expected trend of better performance with decreasing depolarizing noise levels and finite-sampling error (increasing shot count) as can be observed in Figure 2.

## 6.2.2 Advection Equation

To evaluate the framework on a dynamical system, we next consider the 1D advection equation,  $\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} = 0$  for  $x, t \in [0, 1]$ , subject to periodic boundary conditions. Here, the operator learns the solution mapping from the initial condition  $u(x, 0)$  to the full spatiotemporal trajectory  $u(x, t)$ . To strictly enforce the required periodicity, we sample the initial conditions from a Gaussian Random Field utilizing an Exp-Sine-Squared kernel:  $k_l(x_i, x_j) = \exp\left(\frac{-2 \sin^2(\pi d(x_i, x_j)/p)}{l^2}\right)$ . Because advection demands tracking features across spacetime, we deploy a deeper, 7-layer QOrthoNN architecture featuring residual connections to stabilize gradient flow. This model processes 20 uniformly sampled spatial locations for the branch input and a Cartesian product of 50 spatial and temporal coordinates for the trunk query. Consequently, this expanded architecture requires 210 trainable parameters per layer, transpiling to a maximum circuit depth of 305 on IBM Heron and 601 on IBM Eagle. Under ideal simulation, the 8-model advection ensemble successfully maps the dynamics, achieving a 2.28% relative  $L_2$  error and 88.99% empirical coverage for a target rate of  $1 - \alpha = 0.90$ . Having established these theoretical baselines, we now evaluate whether these rigorous conformal coverage guarantees survive under realistic, hardware-calibrated quantum noise.

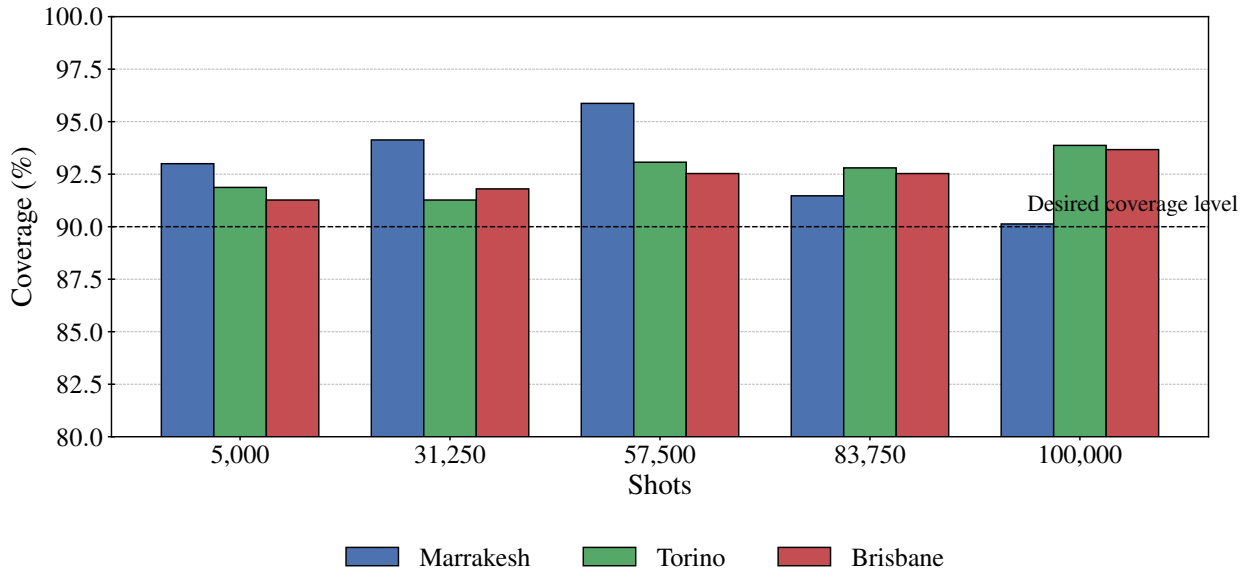


Figure 3: Empirical coverage of the prediction intervals on the test set for the antiderivative task using the complex noise model. We simulated noise models calibrated to IBM Marrakesh, IBM Torino, and IBM Brisbane. The dashed line indicates the target nominal coverage level of 90%. The results demonstrate that the conformal framework maintains validity ( $\geq 90\%$ ) even under the more realistic noise model.

### 6.3 Coverage Under Realistic Noise

To validate practical robustness, we evaluate the empirical coverage of the antiderivative ensemble under realistic noise models calibrated to the IBM Marrakesh, Torino, and Brisbane quantum processors. As illustrated in Figure 3, when evaluated across measurement budgets ranging from 5,000 to 100,000 shots, all three hardware noise profiles successfully maintain empirical coverage at or above the target  $1 - \alpha = 0.90$  threshold. Crucially, this confirms that as long as the hardware error profile—encompassing depolarizing errors, thermal relaxation, and single-qubit readout errors—remains stationary between the calibration and test phases, the inherently noisy quantum execution strictly preserves the exchangeability assumption required for valid conformal prediction. Having demonstrated both sub-quadratic inference and theoretical robustness under physical hardware constraints, we now apply the framework to complex, real-world dynamical systems.

### 6.4 Real Power-Engineering Datasets

Transitioning to real-world power engineering, the complex dynamics of these physical systems impose distinct operational requirements that require specializing our framework into three distinct operator classes. The first is the Transient-to-Transient Operator ( $\mathcal{F}$ ), which maps an entire temporal input trajectory directly to an entire output trajectory. Because this global mapping evaluates the complete dynamic evolution simultaneously, it is essential for offline applications such as post-mortem event analysis and long-term system planning Kwatny & Miu-Miller (2016); Kundur et al. (2004). Formally, this operator functions across the Hilbert spaces  $\mathcal{U} = L^2([0, T]; \mathbb{R}^d)$  and  $\mathcal{Y} = L^2([0, T]; \mathbb{R}^d)$ , where  $d$  represents the state-space dimensionality. Utilizing a dataset of observed input-output trajectory pairs,  $\{(\mathbf{u}_i, \mathbf{s}_i)\}_{i=1}^N$  where  $\mathbf{s}_i = \mathcal{F}[\mathbf{u}_i]$ , the objective is to learn a highly accurate quantum surrogate,  $\hat{\mathcal{F}}$ . We then construct the corresponding pointwise conformal prediction sets,  $C_\alpha(\mathbf{u}_i, t)$  for all  $t \in [0, T]$ , to guarantee reliable inference over the entire simulation horizon.

The second class is the Causal Forecasting Operator ( $\mathcal{H}$ ), which supports proactive grid management strategies such as long-range dynamic stability assessments. Unlike the global mapping of the transient operator,  $\mathcal{H}$  explicitly enforces causality by mapping past pre-event histories to future post-event predictions. Formally defined as  $\mathcal{H} : \mathcal{U}_{\text{past}} \rightarrow \mathcal{Y}_{\text{future}}$ , this operator processes historical input trajectories from

Table 3: Summary of model performance across online and offline experiments under ideal simulation. The target coverage for all uncertainty quantification tasks is 90%.

Experiment	Rel. $L_2$ Error (%)	Coverage (%)	Avg. Width	Peak Uncertainty
Online V-to-V	5.49	90.11	0.148	1.328
Offline V-to-V	12.49	89.66	0.345	2.691
Offline V-to-P	4.08	89.74	0.001	0.063

Table 4: Architecture and hardware specifications for the quantum DeepONet ensembles across online and offline tasks. Width denotes the number of neurons per layer. Params and circuit depth report the maximum per-layer values for trainable RBS angles ( $\theta$ ) and transpiled PQC depth (on IBM Heron/Eagle), respectively. All models use the SiLU activation function. (Config: R = ResNet, F = Fourier features). The number of qubits used for a single model is  $d_u + 2$ .

Experiment	$d_u$	$d_y$	Layers	Width	Params	Heron	Eagle	Config	Ens. Size
Online V-to-V	5	1	2	5	15	94	181	–	4
Offline V-to-V	20	1	6	20	210	305	601	R, F	8
Offline V-to-P	20	1	6	20	210	305	601	R, F	8

the space  $\mathcal{U}_{\text{past}} = L^2([t_0 - \tau, t_0]; \mathbb{R}^d)$  to predict the system’s future dynamic evolution within the space  $\mathcal{Y}_{\text{future}} = L^2([t_0, t_0 + T_f]; \mathbb{R}^d)$ , where  $t_0$  denotes the time of a critical grid contingency and  $d$  represents the physical state-space dimensionality.

The final class is the Causal Point-wise Transient Operator ( $\mathcal{G}$ ), designed for real-time applications such as autonomous grid monitoring and closed-loop control. Mathematically, this is formulated as a time-indexed family of operators,  $\{\mathcal{G}_t\}_{t \in [0, T]}$ . Each specific operator,  $\mathcal{G}_t : L^2([0, t]; \mathbb{R}^n) \rightarrow \mathbb{R}^n$ , maps a historical input trajectory restricted to the current time  $[0, t]$  to a single, localized state prediction at the immediate future step,  $t + \Delta t$ . However, because integrating a continuously expanding historical domain from  $t = 0$  is computationally prohibitive for high-frequency real-time inference, we approximate this mapping using a fixed memory window,  $\tau > 0$ . This ensures operational tractability by truncating the evaluation to a sliding window:  $\mathcal{G}_t[u] \approx \mathcal{M}_t \left[ u(s) \Big|_{s=t-\tau}^t \right]$ . Having formally defined these three specialized operator classes, we now evaluate the framework across a series of highly complex, real-world power engineering tasks.

## 6.5 Offline voltage-to-voltage

We test the Causal Forecasting Operator ( $\mathcal{H}$ ) on an offline voltage prediction task, mapping a 1.9-second pre-fault history to a 2.0-second post-fault trajectory. The input history is discretized into a branch vector  $u \in \mathbb{R}^{20}$ , while the forecast horizon is queried across 100 uniformly spaced trunk coordinates. To mitigate spectral bias caused by severe fault transients, we augment the scalar temporal query,  $t$ , with Fourier features. We extract the  $K = 5$  dominant frequencies via FFT with a Hanning window, mapping  $t$  to an 11-dimensional feature vector:

$$\mathbf{y} = [y, \cos(2\pi f_1 y), \sin(2\pi f_1 y), \dots, \cos(2\pi f_5 y), \sin(2\pi f_5 y)]^\top.$$

To process these signals, we deploy an  $L = 8$  ensemble of 6-layer, 20-neuron QOrthoNNs with residual connections. This configuration requires 210 trainable parameters per layer, transpiling to a maximum quantum circuit depth of 305 on IBM Heron and 601 on IBM Eagle. Under ideal simulation, this framework achieves a 12.49% relative  $L_2$  error. Furthermore, the conformal mechanism yields an 89.66% empirical coverage, effectively maintaining the target threshold of  $1 - \alpha = 0.90$  despite the highly oscillatory dataset.

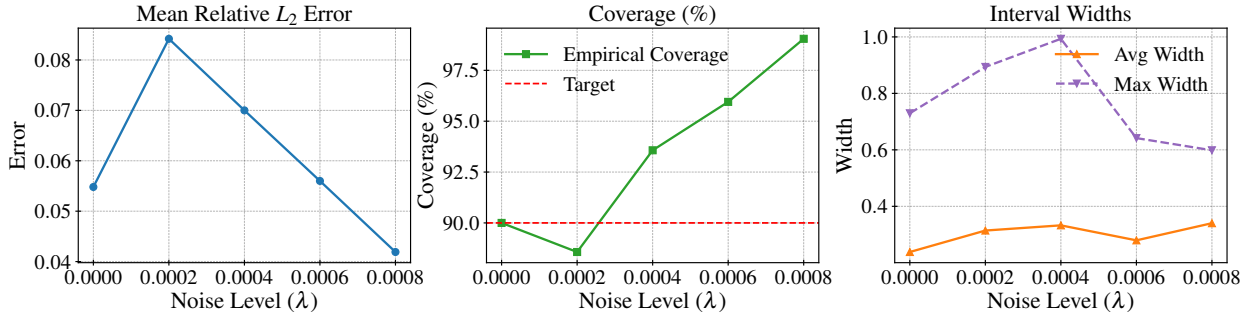


Figure 4: Impact of depolarizing noise  $\lambda$  on Online Voltage-to-Voltage prediction performance (shots=  $10^5$ ). Mean relative  $L_2$  error improves at higher noise levels (left). Empirical coverage also improves at higher noise levels (middle). Evolution of average and maximum prediction interval widths (right).

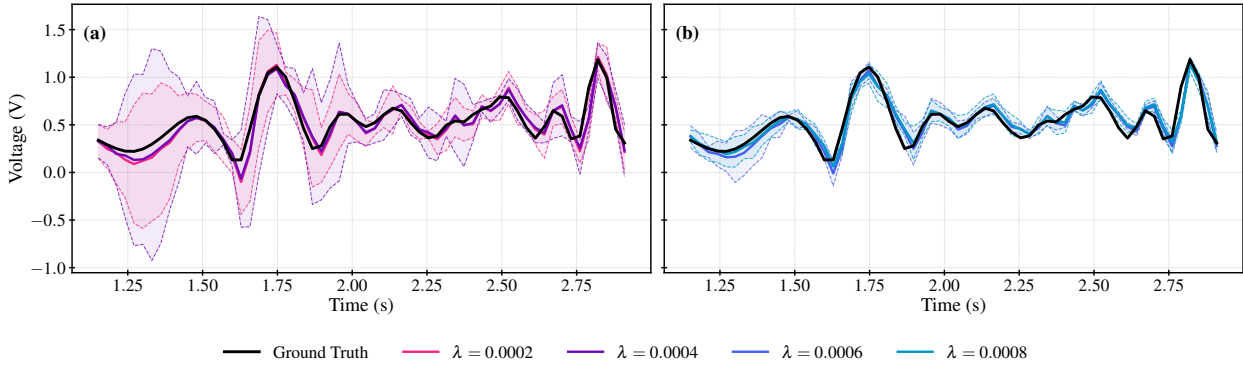


Figure 5: Time-varying tubes constructed by conformal prediction on the online voltage prediction task for a high-variance signal portrayed in different colours to differentiate distinct noise levels. The ground truth is drawn in black and the mean prediction is a solid line of the corresponding colour. Figure (b) shares the same y-axis as figure (a). Interestingly, the average and max width (peak uncertainty) decrease as noise level increases for this particular sample.

## 6.6 Offline voltage-to-active power

We next evaluate the Transient-to-Transient Operator ( $\mathcal{F}$ ) on an offline cross-domain task, mapping a 10-second voltage trajectory to its corresponding active power trajectory. The voltage input is discretized into a branch vector  $v \in \mathbb{R}^{20}$ , while the active power output is queried across 100 Fourier-augmented trunk coordinates. To ensure quantum simulation feasibility against extreme high-frequency transients, the target active power trajectories are smoothed via a zero-phase forward-backward fourth-order Butterworth low-pass filter with a cutoff frequency of  $f_c = f_s/100$ . Thus, the network learns to approximate the filtered operator, formally defined as  $\mathcal{F} : v \mapsto p_{\text{filtered}}$ .

This task deploys the identical  $L = 8$  ensemble architecture used in the prior forecasting benchmark, featuring 6-layer, 20-neuron QOrthoNNs with equivalent parameter counts and quantum circuit depths. Under ideal simulation, this ensemble achieves a 4.08% relative  $L_2$  error. The conformal mechanism again proves robust, delivering an 89.74% empirical coverage against the strict  $1 - \alpha = 0.90$  target threshold.

## 6.7 Online voltage-to-voltage

To conclude the real-world evaluations, our final benchmark tests the Causal Point-wise Transient Operator ( $\mathcal{G}$ ) under a strictly non-exchangeable data structure. We evaluate the Causal Point-wise Transient Operator

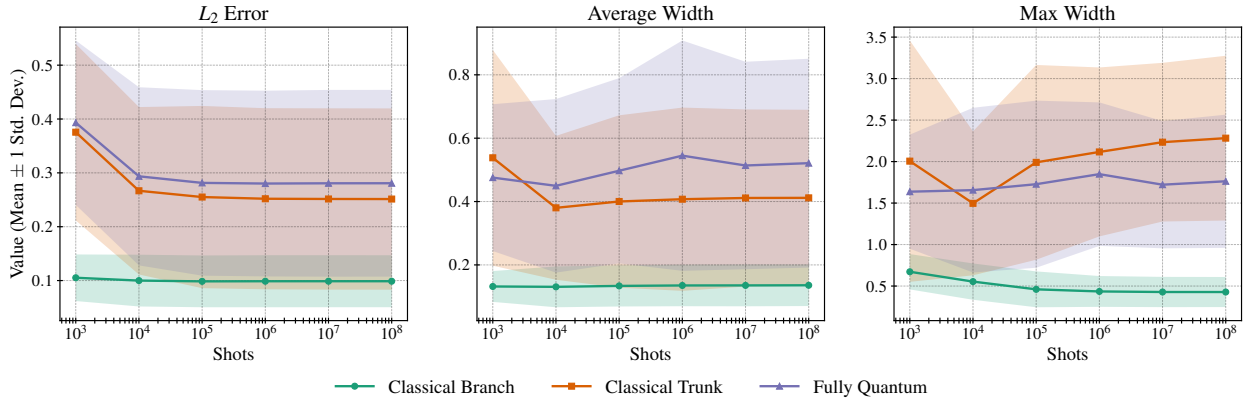


Figure 6: Performance comparison of hybrid classical-quantum architectures on the antiderivative operator task. We evaluate three ensemble configurations: Fully Quantum (purple), Hybrid with Classical Trunk (orange), and Hybrid with Classical Branch (green). The shaded regions indicate  $\pm 1$  standard deviation, where the aggregated predictions were evaluated across a noise parameter space formed by the cross-product of hardware noise (depolarizing levels  $\lambda \in \{0.0002, 0.0004, 0.0006, 0.0008\}$ ) and finite sampling (shot counts ranging from  $10^3$  to  $10^8$ ).

( $\mathcal{G}$ ) on an online voltage prediction task using a sliding memory window of  $\tau = 10$ . For this real-time application, we deploy a compact  $L = 4$  ensemble of 2-layer, 5-neuron QOrthoNNs. This architecture requires only 15 trainable parameters per layer, transpiling to a maximum circuit depth of 94 on IBM Heron. Under ideal simulation, the ensemble achieves a 5.49% relative  $L_2$  error and 90.11% empirical coverage against the  $1 - \alpha = 0.90$  target. Crucially, because the sliding-window data structure introduces strict temporal dependencies, the conformal exchangeability assumption is inherently violated. Thus, this 90.11% result represents a purely empirical observation rather than a distribution-free mathematical guarantee.

Because simulating quantum noise across massive, sequential datasets is computationally intractable, we evaluate the framework’s robustness by applying the simplified depolarizing noise model to targeted data subsamples, including a high-variance, single-signal test set. As illustrated by metrics in Figure 4 and the prediction tubes in Figure 5, these ablations reveal a counterintuitive scaling effect. For shallow circuits using 7 or fewer qubits, desirable uncertainty quantification properties, specifically empirical coverage and average interval width, actually improved at higher depolarizing noise levels ( $\lambda$ ). We discuss the mechanisms driving this anomalous, noise-induced regularization in the limitations section.

It must be noted that the training and evaluation procedures differed between these two figures, leading to distinct manifestations of this effect. For the aggregated metrics (Figure 4), the models were trained and evaluated on randomly subsampled windows drawn from the full signal dataset. In this generalized regime, increased noise improved overall coverage, predictive accuracy, and maximum interval width. Conversely, the visualization (Figure 5) isolates a single high-variance test signal. To satisfy the distributional matching required for conformal prediction to function in this specific qualitative example, the corresponding training set was restricted to high-variance windows. Under these conditions, increased noise primarily drove a significant reduction in the average interval width. The varying expressions of this noise-induced advantage between the two figures are strictly artefacts of these contrasting training and inference procedures.

## 6.8 Hybrid architecture

We isolate the dominant hardware noise source by evaluating fully quantum, classical-branch, and classical-trunk configurations on the antiderivative task. For this specific operator, the quantum branch network is significantly more complex than the trunk, causing it to accumulate substantially more hardware noise and act as the primary fidelity bottleneck. As illustrated in Figure 6, the classical-branch hybrid achieves the lowest relative  $L_2$  error and the narrowest prediction bands across the entire simulated noise space by

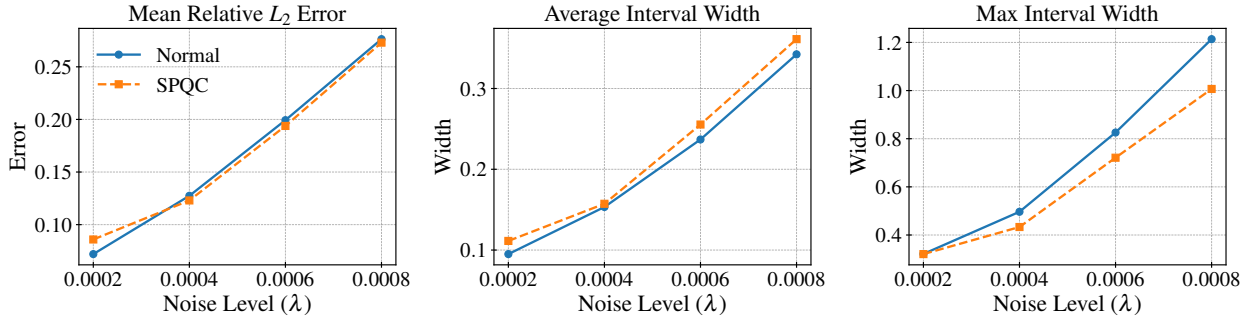


Figure 7: Impact of depolarizing noise  $\lambda$  on the standard ensembling framework and the SPQC architecture (shots= $10^5$ ). The SPQC tracks the standard ensembling framework across all metrics.

Table 5: Circuit resource comparison between normal and SPQC execution modes for a single layer. Width denotes the number of neurons per layer. Circuit depth and gate counts are reported for the transpiled circuits.

Mode	$d_u$	$d_y$	Layers	Width	Depth	CZ	RX	RZ	SX	X
Normal	5	1	2	5	95	62	76	62	21	1
SPQC	5	1	2	5	272	173	144	125	44	1

replacing this bottleneck with a stable classical counterpart. In contrast, the classical-trunk hybrid yields only marginal improvements over the fully quantum baseline because it retains the noisy quantum branch. This confirms that optimal hybrid deployment strictly depends on identifying and replacing the specific sub-network contributing the dominant noise penalty.

## 6.9 Superposed Parameterized Quantum Circuits

We evaluate Superposed Parameterized Quantum Circuits (SPQCs) to determine whether compressing an ensemble into a single unified circuit successfully preserves predictive accuracy and uncertainty quantification. Tested on the antiderivative task using an  $L = 4$  ensemble of 2-layer, 5-neuron QOrthoNNs, the SPQC architecture requires only 7 qubits per sub-network, a strict spatial reduction from the 20 qubits mandated by naïve parallel execution. As illustrated in Figure 7, this unified SPQC strictly matches the standard ensembling baseline across all predictive and uncertainty metrics, maintaining calibration even when subjected to varying levels of depolarizing noise.

Table 5 details the corresponding gate decomposition transpiled onto the IBM Heron processor. While the required count of  $cz$  gates inevitably increases to accommodate the controlled rotations, the counts of the remaining basis gates strictly avoid linear scaling with the ensemble size. Furthermore, pipelining individual RBS gates inherently restrains the maximum SPQC circuit depth to 272. This remains substantially more efficient than the cumulative depth of 380 required to execute four independent, 95-depth standard circuits sequentially. These metrics physically validate the SPQC as a highly scalable, resource-efficient execution strategy whose comparative advantage will strictly compound as hardware coherence improves.

## 7 Limitations and Outlook

Our conformal coverage guarantees strictly rely on data exchangeability. While satisfied in standard batch scenarios, this assumption is inherently violated by temporally dependent streaming data and potentially compromised by correlated quantum noise, such as hardware cross-talk. To safely formalize our empirical

---

online coverage for real-time deployment, future work must extend this framework via adaptive techniques like weighted or sequential conformal prediction. Beyond theoretical calibration, we must physically investigate the anomalous noise-induced regularization observed in shallow circuits to determine if hardware decoherence can be actively harnessed to improve uncertainty bounds in larger models. Finally, subsequent studies must systematically quantify the operational trade-offs between the immediate fidelity advantages of classical-quantum hybrids and the asymptotic resource efficiency of unified SPQCs.

Currently, our empirical validation remains inherently restricted to compact quantum circuits due to the exponential computational overhead of classical simulation. However, as physical hardware coherence inevitably matures, SPQC-based ensembling—driven by its logarithmic spatial scaling—will emerge as the dominant, highly scalable execution strategy. Transitioning this rigorous framework from offline batch analysis to real-time, uncertainty-aware operator learning represents the critical next step toward deploying robust quantum surrogates in complex, safety-critical infrastructure.

## Code Availability

The code supporting the findings of this study is publicly available in the GitHub repository: <https://github.com/purav-0000/conformalized-quantum-deeponet>. The code is released to facilitate reproducibility and further research, and it may be updated over time as improvements are made.

## Acknowledgment

We would like to thank the support of National Science Foundation (DMS-2533878, DMS-2053746, DMS-2134209, ECCS-2328241, CBET-2347401 and OAC-2311848), and U.S. Department of Energy (DOE) Office of Science Advanced Scientific Computing Research program DE-SC0023161, the SciDAC LEADS Institute, and DOE–Fusion Energy Science, under grant number: DE-SC0024583.

## References

- Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: <https://doi.org/10.1016/j.inffus.2021.05.008>. URL <https://www.sciencedirect.com/science/article/pii/S1566253521001081>.
- Muhammad AbuGhanem. Ibm quantum computers: evolution, performance, and future directions. *The Journal of Supercomputing*, 81(5):687, 2025. doi: [10.1007/s11227-025-07047-7](https://doi.org/10.1007/s11227-025-07047-7). URL <https://doi.org/10.1007/s11227-025-07047-7>.
- Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Found. Trends Mach. Learn.*, 16(4):494–591, March 2023. ISSN 1935-8237. doi: [10.1561/2200000101](https://doi.org/10.1561/2200000101). URL <https://doi.org/10.1561/2200000101>.
- Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024. doi: [10.1038/s42254-024-00712-5](https://doi.org/10.1038/s42254-024-00712-5). URL <https://www.nature.com/articles/s42254-024-00712-5>.
- Nathan Baker, Frank Alexander, Timo Bremer, Aric Hagberg, Yannis Kevrekidis, Habib Najm, Manish Parashar, Abani Patra, James Sethian, Stefan Wild, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Technical report, USDOE Office of Science (SC), Washington, DC, USA, 2019.
- Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermann Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Rev. Mod.*

- 
- Phys.*, 94:015004, Feb 2022. doi: 10.1103/RevModPhys.94.015004. URL <https://link.aps.org/doi/10.1103/RevModPhys.94.015004>.
- Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113.
- John C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 3rd edition, 2016.
- Germund Dahlquist and Åke Björck. *Numerical Methods in Scientific Computing, Volume I*. SIAM, 2008.
- Afrah Farea, Saiful Khan, and M Serdar Çelebi. Qcpinn: Quantum-classical physics-informed neural networks for solving pdes. *Machine Learning: Science and Technology*, 6(4):045053, 2025. doi: 10.1088/2632-2153/aelc91. URL <https://iopscience.iop.org/article/10.1088/2632-2153/aelc91>.
- Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 48: 1050–1059, 2016. URL <https://proceedings.mlr.press/v48/gal16.html>.
- Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, pp. 45–87. Springer International Publishing, Cham, 2020. ISBN 978-3-030-42553-1. doi: 10.1007/978-3-030-42553-1\_3. URL [https://doi.org/10.1007/978-3-030-42553-1\\_3](https://doi.org/10.1007/978-3-030-42553-1_3).
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2016. ISBN 9780262035613. URL <https://www.deeplearningbook.org>.
- Bin Huang and Jianhui Wang. Applications of physics-informed neural networks in power systems — a review. *IEEE Transactions on Power Systems*, 38(1):572–588, 2023. doi: 10.1109/TPWRS.2022.3162473.
- IBM Quantum. Quantum computers. <https://quantum.cloud.ibm.com/computers>, 2026. Accessed: 2026-04-13.
- Bruno Jacob, Amanda A. Howard, and Panos Stinis. Bridging quantum and classical computing for partial differential equations through multifidelity machine learning, 2025. URL <https://arxiv.org/abs/2512.05241>.
- Nishant Jain, Jonas Landman, Natansh Mathur, and Iordanis Kerenidis. Quantum fourier networks for solving parametric pdes. *Quantum Science and Technology*, 9(3):035026, 2024. doi: 10.1088/2058-9565/ad42ce.
- Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit, 2024.
- Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *arXiv preprint arXiv:2011.04520*, 2020.
- Kui Jia, Shuai Li, Yuxin Wen, Tongliang Liu, and Dacheng Tao. Orthogonal deep neural networks, 2019. URL <https://arxiv.org/abs/1905.05929>.
- Jaewoo Joo and Hyungil Moon. Quantum variational PDE solver with machine learning. *arXiv preprint arXiv:2109.09216*, 2021.
- George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.
- Iordanis Kerenidis, Jonas Landman, and Natansh Mathur. Classical and quantum algorithms for orthogonal neural networks, 2022. URL <https://arxiv.org/abs/2106.07198>.

- 
- P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziaargyriou, D. Hill, A. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal. Definition and classification of power system stability ieeecigre joint task force on stability terms and definitions. *IEEE Transactions on Power Systems*, 19(3):1387–1401, 2004. doi: 10.1109/TPWRS.2004.825981.
- Harry G. Kwatny and Karen Miu-Miller. *Power System Dynamics and Control*. Control Engineering. Springer, New York, NY, 2016. ISBN 978-0-8176-4674-5. doi: 10.1007/978-0-8176-4674-5. URL <https://doi.org/10.1007/978-0-8176-4674-5>.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*, pp. 6405–6416, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html>.
- Jonas Landman, Natansh Mathur, Yun Yvonna Li, Martin Strahm, Skander Kazdaghi, Anupam Prakash, and Iordanis Kerenidis. Quantum methods for neural networks and application to medical image classification. *Quantum*, 6:881, dec 2022. ISSN 2521-327X. doi: 10.22331/q-2022-12-22-881. URL <https://quantum-journal.org/papers/q-2022-12-22-881/>.
- Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David J. Crandall, and Dhruv Batra. Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint*, arXiv:1511.06314, 2015. URL <https://arxiv.org/abs/1511.06314>.
- Wing Tat Leung, Guang Lin, and Zecheng Zhang. NH-PINN: Neural homogenization-based physics-informed neural network for multiscale problems. *Journal of Computational Physics*, 470:111539, 2022. doi: 10.1016/j.jcp.2022.111539.
- Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL <https://www.jmlr.org/papers/v24/21-1524.html>.
- Guang Lin, Christian Moya, and Zecheng Zhang. Learning the dynamical response of nonlinear non-autonomous dynamical systems with deep operator neural networks. *Engineering Applications of Artificial Intelligence*, 125:106689, 2023. ISSN 0952-1976. doi: <https://doi.org/10.1016/j.engappai.2023.106689>. URL <https://www.sciencedirect.com/science/article/pii/S0952197623008734>.
- Jiayu Liu, Jun Liu, Jie Zhang, Wanliang Fang, and Longchu Qu. Power system stochastic transient stability assessment based on monte carlo simulation. *The Journal of Engineering*, 2019(16):1051–1055, 2019. doi: <https://doi.org/10.1049/joe.2018.8471>. URL <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2018.8471>.
- Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL <https://doi.org/10.1038/s42256-021-00302-5>.
- Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022a. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2022.114778>. URL <https://www.sciencedirect.com/science/article/pii/S0045782522001207>.
- Lu Lu, Raphaël Pestourie, Steven G. Johnson, and Giuseppe Romano. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Phys. Rev. Res.*, 4:023210, Jun 2022b. doi: 10.1103/PhysRevResearch.4.023210. URL <https://link.aps.org/doi/10.1103/PhysRevResearch.4.023210>.

- 
- David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.
- Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018. doi: 10.1038/s41467-018-07090-4. URL <https://www.nature.com/articles/s41467-018-07090-4>.
- Jovica V. Milanović. Probabilistic stability analysis: the way forward for stability analysis of sustainable power systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2100):20160296, 07 2017. ISSN 1364-503X. doi: 10.1098/rsta.2016.0296. URL <https://doi.org/10.1098/rsta.2016.0296>.
- Christian Moya and Guang Lin. DAE-PINN: A physics-informed neural network model for simulating differential algebraic equations with application to power networks. *Neural Computing and Applications*, 35(5):3789–3804, 2023. doi: 10.1007/s00521-022-07886-y.
- Christian Moya, Shiqi Zhang, Guang Lin, and Meng Yue. Deepnet-grid-uq: A trustworthy deep operator framework for predicting the power grid’s post-fault trajectories. *Neurocomputing*, 535:166–182, 2023. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2023.03.015>. URL <https://www.sciencedirect.com/science/article/pii/S0925231223002503>.
- Christian Moya, Amirhossein Mollaali, Zecheng Zhang, Lu Lu, and Guang Lin. Conformalized-deepnet: A distribution-free framework for uncertainty quantification in deep operator networks. *Physica D: Nonlinear Phenomena*, 471:134418, 2025. ISSN 0167-2789. doi: <https://doi.org/10.1016/j.physd.2024.134418>. URL <https://www.sciencedirect.com/science/article/pii/S0167278924003683>.
- Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0. URL <https://doi.org/10.1007/978-1-4612-0745-0>.
- Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen (eds.), *Machine Learning: ECML 2002*, pp. 345–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-36755-0.
- Sangwoo Park and Osvaldo Simeone. Quantum conformal prediction for reliable uncertainty quantification in quantum machine learning. *IEEE Transactions on Quantum Engineering*, 5:1–24, 2024. doi: 10.1109/TQE.2023.3333224.
- Viktoria Patapovich, Mo Kordzanganeh, and Alexey Melnikov. Superposed parameterised quantum circuits. *arXiv preprint*, arXiv:2506.08749, 2025. URL <https://arxiv.org/abs/2506.08749>. arXiv:2506.08749v1 [quant-ph].
- John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. ISSN 2521-327X. doi: 10.22331/q-2018-08-06-79. URL <https://doi.org/10.22331/q-2018-08-06-79>.
- Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2022.111902>. URL <https://www.sciencedirect.com/science/article/pii/S0021999122009652>.
- Tong Qin, Kailiang Wu, and Dongbin Xiu. Data-driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019. doi: 10.1016/j.jcp.2019.06.042.
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.
- Maziar Raissi, Paris Perdikaris, and George Em 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. doi: 10.1016/j.jcp.2018.10.045.

- 
- Marco Schumann, Frank K. Wilhelm, and Alessandro Ciani. Emergence of noise-induced barren plateaus in arbitrary layered noise models. *Quantum Science and Technology*, 9(4):045019, 2024. doi: 10.1088/2058-9565/ad6285. URL <https://doi.org/10.1088/2058-9565/ad6285>.
- Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008. URL <https://www.jmlr.org/papers/volume9/shafer08a/shafer08a.pdf>.
- Corey Trahan, Mark Loveland, and Samuel Dent. Quantum physics-informed neural networks. *Entropy*, 26(8):649, 2024. doi: 10.3390/e26080649.
- Vladimir Vovk, Alexander Gammernan, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer Series in Statistics. Springer, Cham, 2 edition, 2022. ISBN 978-3-031-06649-8. doi: 10.1007/978-3-031-06649-8. URL <https://doi.org/10.1007/978-3-031-06649-8>.
- Volodya Vovk, Alexander Gammernan, and Craig Saunders. Machine-learning applications of algorithmic randomness. In *Proceedings of the Sixteenth International Conference on Machine Learning, ICML '99*, pp. 444–453, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.
- Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12:6961, 2021. doi: 10.1038/s41467-021-27045-6. URL <https://www.nature.com/articles/s41467-021-27045-6>.
- Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022. ISSN 0309-1708. doi: <https://doi.org/10.1016/j.advwatres.2022.104180>. URL <https://www.sciencedirect.com/science/article/pii/S0309170822000562>.
- Nick Winovich, Karthik Ramani, and Guang Lin. ConvPDE-UQ: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains. *Journal of Computational Physics*, 394:263–279, 2019. doi: 10.1016/j.jcp.2019.05.026.
- Pengpeng Xiao, Muqing Zheng, Anran Jiao, Xiu Yang, and Lu Lu. Quantum DeepONet: Neural operators accelerated by quantum computing. *Quantum*, 9:1761, June 2025. ISSN 2521-327X. doi: 10.22331/q-2025-06-04-1761. URL <https://doi.org/10.22331/q-2025-06-04-1761>.
- Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021. doi: 10.1016/j.jcp.2020.109913.
- Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS Computational Biology*, 16(11):e1007575, 2020. doi: 10.1371/journal.pcbi.1007575.
- Ketian Ye, Junbo Zhao, Nan Duan, and Yingchen Zhang. Physics-informed sparse gaussian process for probabilistic stability analysis of large-scale power system with dynamic pvs and loads. *IEEE Transactions on Power Systems*, 38(3):2868–2879, 2023. doi: 10.1109/TPWRS.2022.3188182.
- Jiaxin Zhang. Modern monte carlo methods for efficient uncertainty quantification and propagation: A survey. *WIREs Computational Statistics*, 13(5):e1539, 2021. doi: <https://doi.org/10.1002/wics.1539>. URL <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1539>.
- Zongren Zou, Xuhui Meng, Apostolos F. Psaros, and George E. Karniadakis. Neuraluq: A comprehensive library for uncertainty quantification in neural differential equations and operators. *SIAM Review*, 66(1): 161–190, 2024. doi: 10.1137/22M1518189. URL <https://doi.org/10.1137/22M1518189>.

---

## Appendix

### A Detailed QOrthoNN quantum state evolution

For a general  $m \times n$  transformation  $\mathbf{W}$ , the required number of data qubits is  $q = \max(m, n)$ . Let  $|\mathbf{e}_k\rangle$  denote the physical unary basis state where the  $k$ -th qubit is in the  $|1\rangle$  state.

1. (a) If  $m = n$ , then  $q = n$  and all data qubits are used at every stage of the circuit.  
 (b) If  $m > n$ , then  $q = m$ . The classical data is encoded onto only the bottom  $n$  qubits (indices  $m - n + 1$  to  $m$ ), and all  $m$  qubits are measured.  
 (c) If  $m < n$ , then  $q = n$ . The classical data is encoded on all  $n$  qubits (indices 1 to  $n$ ), and only the bottom  $m$  qubits (indices  $n - m + 1$  to  $n$ ) are measured.
2. **State preparation** — For an  $n$ -dimensional input vector  $\mathbf{x}$  and an  $m \times n$  linear transformation,  $q$  qubits are initialized in the ground state along with an ancilla qubit in uniform superposition. The ancilla qubit is used to infer the sign of the magnitude of the output vector since a quantum measurement natively yields positive probabilities.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle^{\otimes q} + \frac{1}{\sqrt{2}}|1\rangle|0\rangle^{\otimes q}$$

Where the first register corresponds to the ancilla qubit and the second register corresponds to the data qubits.

A CNOT gate controlled by the ancilla qubit acts on the first active input qubit (index  $q - n + 1$ ) of the data register, creating entanglement and the resultant state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle^{\otimes q} + \frac{1}{\sqrt{2}}|1\rangle|\mathbf{e}_{q-n+1}\rangle$$

where  $|\mathbf{e}_{q-n+1}\rangle$  is a unary vector of size  $q$  with the  $(q - n + 1)$ -th qubit set to  $|1\rangle$ .

3. **Amplitude encoding in a unary subspace** — Given an input vector  $\mathbf{x}$  and the qubits initialized in the unary state  $|\mathbf{e}_{q-n+1}\rangle$ , we encode  $\mathbf{x}$  in the Hilbert space as:

$$S(\mathbf{x})|\mathbf{e}_{q-n+1}\rangle = x_1|\mathbf{e}_{q-n+1}\rangle + x_2|\mathbf{e}_{q-n+2}\rangle + \cdots + x_n|\mathbf{e}_q\rangle = \sum_{i=1}^n x_i|\mathbf{e}_{q-n+i}\rangle$$

where  $S(\mathbf{x})$  is a data-dependent encoding unitary operating on the bottom  $n$  qubits.

Because the unitary  $S(\mathbf{x})$  is constructed entirely from Reconfigurable Beam Splitter (RBS) gates, which preserve the  $|00\rangle$  state, it has no effect on the ground state ( $S(\mathbf{x})|0\rangle^{\otimes q} = |0\rangle^{\otimes q}$ ). Consequently, applying the operator  $I \otimes S(\mathbf{x})$  to the full system transforms the overall state to:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle^{\otimes q} + \frac{1}{\sqrt{2}}|1\rangle \sum_{i=1}^n x_i|\mathbf{e}_{q-n+i}\rangle$$

4. **Pyramidal orthogonal transformation** — Any orthogonal matrix  $\mathbf{W}$  in  $\mathbf{y} = \mathbf{W}\mathbf{x}$  can be decomposed into a network of RBS gates under nearest-neighbour connectivity. This yields a parameterized unitary  $U(\boldsymbol{\theta})$  that maps the  $n$ -dimensional input subspace to the  $m$ -dimensional output subspace:

$$U(\boldsymbol{\theta})S(\mathbf{x})|\mathbf{e}_{q-n+1}\rangle = \sum_{j=1}^m \left( \sum_{i=1}^n W_{ji}x_i \right) |\mathbf{e}_{q-m+j}\rangle$$

Because the parameterized unitary  $U(\boldsymbol{\theta})$  is also constructed entirely from RBS gates, it preserves the ground state ( $U(\boldsymbol{\theta})|0\rangle^{\otimes q} = |0\rangle^{\otimes q}$ ). Applying the operator  $I \otimes U(\boldsymbol{\theta})$  to the joint system yields the overall state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle^{\otimes q} + \frac{1}{\sqrt{2}}|1\rangle \sum_{j=1}^m \left( \sum_{i=1}^n W_{ji}x_i \right) |\mathbf{e}_{q-m+j}\rangle$$

5. **Tomography and sign extraction** — Finally, we must recover the transformed components  $\sum_{i=1}^n W_{ji}x_i$  from the circuit. To do so, we introduce  $S(\mathbf{r})$ , an encoding unitary parameterized to load the uniform norm-1 vector  $\mathbf{r} = (\frac{1}{\sqrt{q}}, \frac{1}{\sqrt{q}}, \dots, \frac{1}{\sqrt{q}})$  onto the full  $q$ -dimensional subspace. First, we apply the adjoint operator  $I \otimes S^\dagger(\mathbf{r})$  to the full system.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle^{\otimes q} + \frac{1}{\sqrt{2}}|1\rangle S^\dagger(\mathbf{r}) \sum_{j=1}^m \left( \sum_{i=1}^n W_{ji}x_i \right) |\mathbf{e}_{q-m+j}\rangle$$

Second, we flip the ancilla qubit with an X gate and perform a CNOT gate on the first active qubit (index 1) of the data register, controlled by the ancilla qubit.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|1\rangle|\mathbf{e}_1\rangle + \frac{1}{\sqrt{2}}|0\rangle S^\dagger(\mathbf{r}) \sum_{j=1}^m \left( \sum_{i=1}^n W_{ji}x_i \right) |\mathbf{e}_{q-m+j}\rangle$$

Third, we apply  $I \otimes S(\mathbf{r})$  to the full system. On the  $|1\rangle$  branch, this loads the uniform vector  $\mathbf{r}$ . On the  $|0\rangle$  branch,  $S(\mathbf{r})$  and  $S^\dagger(\mathbf{r})$  perfectly cancel out ( $S(\mathbf{r})S^\dagger(\mathbf{r}) = I$ ), restoring the transformed data components.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|1\rangle \sum_{k=1}^q \frac{1}{\sqrt{q}}|\mathbf{e}_k\rangle + \frac{1}{\sqrt{2}}|0\rangle \sum_{j=1}^m \left( \sum_{i=1}^n W_{ji}x_i \right) |\mathbf{e}_{q-m+j}\rangle$$

Finally, a Hadamard gate on the ancilla qubit mixes the states to provide the result:

$$|\psi\rangle = \frac{1}{2}|0\rangle \left( \sum_{j=1}^m y_j |\mathbf{e}_{q-m+j}\rangle + \sum_{k=1}^q \frac{1}{\sqrt{q}} |\mathbf{e}_k\rangle \right) + \frac{1}{2}|1\rangle \left( \sum_{j=1}^m y_j |\mathbf{e}_{q-m+j}\rangle - \sum_{k=1}^q \frac{1}{\sqrt{q}} |\mathbf{e}_k\rangle \right)$$

where  $y_j = \sum_{i=1}^n W_{ji}x_i$ . By isolating the measurements on the specific  $m$ -dimensional subspace, the  $j$ -th component of the output vector can then be inferred using the formula:

$$\sqrt{q}(\Pr[0, \mathbf{e}_{q-m+j}] - \Pr[1, \mathbf{e}_{q-m+j}]) = \frac{\sqrt{q}}{4} \left( y_j + \frac{1}{\sqrt{q}} \right)^2 - \frac{\sqrt{q}}{4} \left( y_j - \frac{1}{\sqrt{q}} \right)^2 = y_j$$

---

## B Nomenclature

### Data and Operator Formulation

$\mathcal{G}$  Target operator mapping input functions to pointwise predictions.

$\mathcal{D}$  Supervised training dataset.

$\mathbf{u}_i, \mathbf{u}$  Discretized input function evaluated at fixed sensor locations.

$d_u$  Dimensionality of the discretized input function.

$\mathbf{y}_{ij}, \mathbf{y}$  Spatial or temporal query coordinate.

$d_y$  Dimensionality of the query coordinate.

$s_{ij}$  Ground-truth value of the operator evaluated at a specific coordinate.

$N$  Total number of input trajectories or hypothetical scenarios in a batch.

$M, M_i$  Maximum and specific number of spatial/temporal query coordinates evaluated per scenario.

### Network Architecture

$\mathbf{b}$   $p$ -dimensional branch network output vector.

$\mathbf{t}$   $p$ -dimensional trunk network output vector.

$\theta$  Trainable parameters of the sub-networks, representing Reconfigurable Beam Splitter rotation angles.

$d_\theta$  Total number of trainable parameters.

$n$  Hidden layer width and input vector dimension for orthogonal linear transformations.

$\mathbf{W}$  Orthogonal weight matrix emulated by the parameterized quantum circuit.

### Quantum Circuit

$L$  Number of independent quantum DeepONet models in the ensemble.

$S(\cdot)$  Data loader parameterized unitary circuit utilizing amplitude encoding.

$U(\cdot)$  Pyramidal parameterized unitary circuit executing orthogonal linear transformations.

$|\mathbf{e}_i\rangle$  Unary computational basis state within the quantum register.

$N_{shots}$  Total number of quantum measurement shots budgeted per inference.

$\lambda$  Noise strength parameter for simulated depolarizing channels.

### Uncertainty Quantification

$\mu(\mathbf{u})(\mathbf{y})$  Mean prediction output calculated across the ensemble.

$\sigma(\mathbf{u})(\mathbf{y})$  Ensemble standard deviation quantifying epistemic disagreement.

$r_{ij}$  Adaptive nonconformity score evaluating prediction error normalized by local variance.

$\hat{q}$  Empirical quantile of the nonconformity scores derived from calibration data.

$C_\alpha(\mathbf{u}, \mathbf{y})$  Adaptive, distribution-free prediction set (safety tube).

$\alpha$  User-specified miscoverage rate for the conformal prediction intervals.

Table 6: Training hyperparameters. “Full” indicates full-batch gradient descent.  $\gamma$  is the decay factor. “–” indicates a constant learning rate (no scheduler).

Experiment	$N_{train}$	$N_{cal}$	$N_{test}$	Batch	Iters	LR	Min LR	$\gamma$	Loss
Antiderivative	200	50	50	Full	30k	1e-3	–	–	MSE
Advection	1000	200	200	Full	40k	1e-3	5e-4	0.99	MSE
Offline V-to-V	0.8	0.1	0.1	256	40k	5e-4	–	–	MSE
Offline V-to-P	0.8	0.1	0.1	64	40k	5e-3	5e-4	0.99	Rel. $L_2$
Online V-to-V	0.8	0.1	0.1	256	15k	1e-2	–	–	MSE

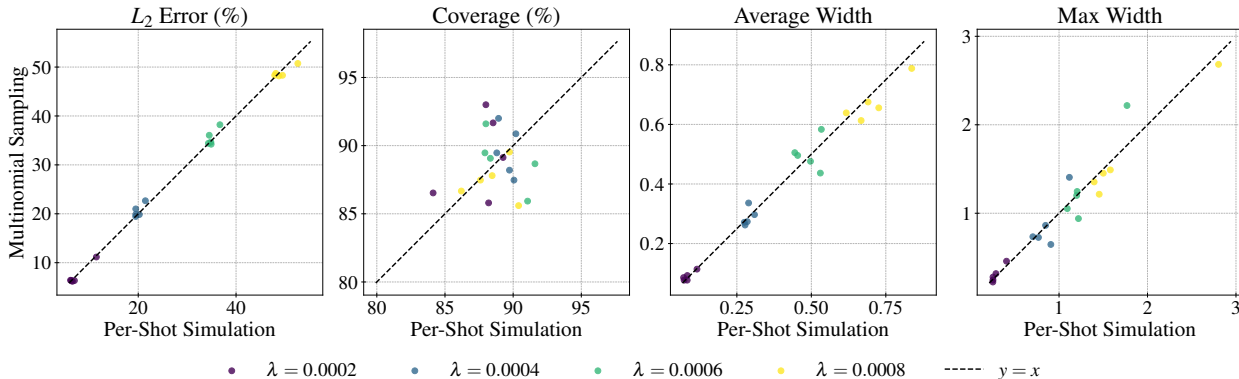


Figure 8: Validation of multinomial sampling. Parity plots comparing the outputs of per-shot simulation (x-axis) against multinomial sampling (y-axis). The dashed line represents  $y = x$ .

## C Training hyperparameters for Ideal Simulations

We list the hyperparameters used during training in 6. We utilized Lambda decay to reduce the learning rate.

## D Multinomial sampling vs per-shot simulations

To accelerate the simulation of quantum circuits, we employ multinomial sampling from the final statevector or density matrix probability distribution, rather than simulating each shot individually. This single simulation followed by sampling is computationally efficient.

To validate this approach, we compare its results against a full per-shot simulation, which applies noise at every step for every shot. Figure 8 shows this comparison for the antiderivative operator. The results for all metrics (L2 Error, Coverage, Average Width, and Max Width) demonstrate correlation, validating that our multinomial sampling method is an accurate and efficient substitute for the more costly per-shot simulation.

## E System snapshots

The realistic noise model (Section 5.2.1) was constructed based on calibration data from three Qiskit backends: `fake_brisbane`, `fake_torino`, and `fake_marrakesh`. The noise model was generated using these properties. All simulations were transpiled to the basis gate set and coupling map of the respective backend.

Table 7: Backend characteristics for the simulated quantum devices. Reported values include hardware configuration, coherence times ( $T_1$ ,  $T_2$ ), readout errors, and gate errors. All values are averaged, minimum, and maximum across qubits unless otherwise specified.

Backend	Qubits	Basis Gates	Coupling
fake_brisbane	127	{ecr, id, rz, sx, x}	144 edges
fake_torino	133	{cz, id, rz, sx, x}	300 edges
fake_marrakesh	156	{cz, id, rz, sx, x}	352 edges

Backend	$T_1$ (avg)	$T_1$ (min)	$T_1$ (max)	$T_2$ (avg)	$T_2$ (min)	$T_2$ (max)
fake_brisbane	2.34e8	9.94e6	4.25e8	1.61e8	1.71e7	4.12e8
fake_torino	1.74e8	3.16e6	3.08e8	1.45e8	6.13e6	3.50e8
fake_marrakesh	2.07e8	6.80e6	4.99e8	1.47e8	1.80e5	5.35e8

Backend	Readout Err (avg)	Readout Err (min)	Readout Err (max)
fake_brisbane	3.11	0.56	23.95
fake_torino	4.67	0.51	56.52
fake_marrakesh	2.70	0.27	47.39

Backend	1Q Err (avg)	1Q Err (min)	1Q Err (max)	2Q Err (avg)
fake_brisbane	0.11	0.00	14.09	1.96
fake_torino	0.06	0.00	1.74	8.00
fake_marrakesh	0.06	0.00	4.20	8.06