Quantum Neural Operators: Implicit Quadratic Frame and Expressivity Advantages with Adaptive Frequency Strategy

Ruochen Wang IGNITE@SJTU.EDU.CN
School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, 200030, China
Shanghai Innovation Institute, Shanghai, 200003, China

Zhuo Xia CHIRTEE.XIAZHUO@SJTU.EDU.CN
School of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China

Xiaoqiu Zhong 21-ZXQ@SJTU.EDU.CN
School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, 200030, China
Shanghai Innovation Institute, Shanghai, 200003, China

Junchi Yan[*] YANJUNCHI@SJTU.EDU.CN
School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, 200030, China
Shanghai Innovation Institute, Shanghai, 200003, China

Abstract

While neural operators have advanced scientific machine learning, developing efficient quantum counterparts remains challenging due to hybrid processing overheads and theoretical gaps. In this work, we propose QuanONet, a pure quantum neural operator framework tailored for the Noisy Intermediate-Scale Quantum (NISQ) era. Theoretically, we extend the universal approximation for operators to quantum computing, rigorously proving QuanONet's capability to approximate continuous nonlinear operators. Departing from conventional views attributing quantum advantage solely to exponential Hilbert space dimensions, we demonstrate that the architecture implicitly constructs an effective frame of size $\mathcal{O}(p^2)$ via the density matrix, thereby exceeding the representational capacity of classical models with the equivalent latent dimension p. Crucially, we identify that vanilla Fixed-Frequency embeddings induce dimensional collapse of the structural frame. To address this, we introduce the adaptive Trainable-Frequency (TF-QuanONet) strategy, which serves as a structural necessity to unlock the $\mathcal{O}(p^2)$ expressivity. Extensive experiments on benchmarks including the Antiderivative operator and Diffusion-Reaction system demonstrate that TF-QuanONet significantly outperforms quantum baselines. It also surpasses state-of-the-art classical frameworks under most stress tests where classical models utilize nearly an order of magnitude more parameters. Furthermore, in high-dimensional scaling regimes ($p \to 256$), TF-QuanONet demonstrates superior optimization robustness, strictly converging to the intrinsic error floor while classical baselines suffer from stochastic instability despite massive parameterization. The deployment on IBM quantum processors validates its functional viability on NISQ hardware and near-term applicability.

Keywords: Neural Operator, Quantum Neural Networks, Differential Equations, Quantum Advantages

∗. Correspondence author.

1 Introduction

As a potentially disruptive technology, quantum computing has been intensively studied and further developed for solving partial differential equations (PDEs) (Montanaro and Pallister, 2016; Arrazola et al., 2019; Lloyd et al., 2020; Liu et al., 2023). The primary objective is to establish new paradigms for PDE solving that achieve quantum advantages in either theoretical complexity or empirical efficiency. One prominent advancement is the Schrödingerization technique proposed by Jin et al. (2024), which mathematically demonstrates how to achieve exponential complexity reduction over classical methods (Jin et al., 2025). Parallel to this, numerous quantum variational methods and their circuit embodiments have been developed (Kyriienko et al., 2021; Xiao et al., 2024b), with applications demonstrated in fluid dynamics and engineering (Azizzadenesheli et al., 2024). However, data-driven machine learning paradigms have rarely been integrated into these quantum solvers.

Conversely, machine learning has been extensively applied to PDE solving over the past decade (Raissi et al., 2019; Lu et al., 2021; Karniadakis et al., 2021). These approaches are generally categorized into two streams: neural operator learning and frequency domain transformation. They typically perceive the input function state through a finite set of m discrete sensors, learning to approximate the operator mapping from these local observations. Grounded in the universal approximation theory for operators (Chen and Chen, 1995; Kovachki et al., 2021), these methods have achieved empirical success due to their fast inference capabilities (Li et al., 2021; Wen et al., 2023). Nevertheless, in complex physical scenarios requiring high-dimensional representations, classical learning-based methods often suffer from optimization instability and performance saturation due to their linear frame limitations.

Motivated by these complementary advancements and their respective limitations, this paper explores the intersection of these two domains. Technically, achieving effective quantum machine learning for PDEs—especially with concrete, native quantum circuits and justifiable theoretical support—remains an open and intellectually challenging question. This exploration is particularly pertinent given that quantum machines can, in certain regimes, exhibit superior generalization abilities compared to classical counterparts (Abbas et al., 2021; Caro et al., 2022). Furthermore, we aim to expand the frontier of Quantum Machine Learning (QML) beyond traditional data formats. Given that surpassing classical benchmarks in text and image processing remains difficult for QML in the NISQ era (Preskill, 2018; Cerezo et al., 2022), scientific computing represents a promising avenue for demonstrating practical quantum utility.

To this end, we present QuanONet, a quantum neural operator framework with theoretical guarantees and implementations on both simulators and real quantum processors. As a general framework for learning nonlinear operators, QuanONet is designed to be hardware-efficient, featuring logarithmic qubit scaling $\mathcal{O}(\log p)$ and minimal connectivity requirements, making it feasible for NISQ devices while holding potential for future fault-tolerant systems.

Theoretically, QuanONet is grounded in our extension of the universal approximation theorem to quantum version. Specifically, we formulate this theorem by the physical measurement of expectation values—mathematically represented as the Frobenius inner product

of a modulated observable against a density matrix. We reveal that the model's superior expressivity stems from the quadratic structure of the quantum density matrix: although parameterized by a latent dimension of $\mathcal{O}(p)$, the architecture implicitly constructs an effective frame of size $\mathcal{O}(p^2)$ via the density matrix. This allows QuanONet to surpass the representational capacity of classical models restricted to linear frames. Crucially, this gain in expressivity does not induce overfitting; rather, the compact parameterization inherent to the quantum circuit imposes a strong inductive bias and geometric regularization. This results in not only a minimal generalization gap but also superior optimization stability in high-dimensional parameter spaces, compared to the stochastic instability often observed in classical baselines.

Furthermore, we identify that standard Fixed-Frequency embeddings lead to a dimensional collapse of this implicit frame due to spectral symmetries. To resolve this, we incorporate the Trainable-Frequency (TF-QuanONet) strategy (Jaderberg et al., 2024). By treating embedding coefficients as learnable parameters, this method ensures spectral distinction, which is mathematically requisite to break the linear dependency of the frame functions and realize the $\mathcal{O}(p^2)$ representational capacity of the density matrix.

The contributions of this paper are summarized as follows[1]:

1) We extend the universal approximation theorem to the quantum domain (Theorem 3), proving that quantum measurement constructs an implicit quadratic frame of size $\mathcal{O}(p^2)$. This structure offers a fundamental representational advantage over classical linear frames of size $\mathcal{O}(p)$ given the same latent dimension. To our knowledge, this is the first theoretical justification for Quantum Neural Operators, bridging the gap left by previous studies confined to target function learning (Schuld et al., 2021; Gonon and Jacquier, 2025).

2) We propose QuanONet, a general framework for learning nonlinear operators using pure quantum circuits. In contrast to hybrid architectures like QDeepONet (Xiao et al., 2024a), our design eliminates classical-quantum communication bottlenecks during inference. By featuring logarithmic qubit scaling and minimal connectivity requirements, QuanONet achieves high hardware efficiency, ensuring practical feasibility for deployment on NISQ processors.

3) We introduce the Trainable-Frequency (TF-QuanONet) embedding strategy to optimize spectral representation. We demonstrate that this mechanism serves as a mathematical prerequisite, rather than a mere optimization trick, to prevent the dimensional collapse induced by Fixed-Frequency grids. By ensuring spectral distinction, this strategy effectively unlocks the full $\mathcal{O}(p^2)$ representational capacity inherent to the density matrix structure.

4) We conduct extensive experiments extending to high-dimensional latent spaces ($p = 256$), demonstrating a distinct quantum advantage. TF-QuanONet not only outper-

1. This is an extended version of the preliminary results in the conference version (Wang et al., 2025), improved with: 1) new benchmarks (Advection/Darcy flow); 2) proposing and giving theoretical proofs for the implicit quadratic frame; 3) comprehensive experimental evaluations involving asymmetric stress tests, extended analysis on high-dimensional scaling limits, and detailed ablation studies on circuit architecture and spectral analysis; 4) the paper is heavily rewritten and extended with more details.

forms classical baselines (DeepONet and FNO) in asymmetric stress tests but also exhibits superior optimization stability. While classical models suffer from significant stochastic variance in high dimensions, our quantum framework maintains consistent convergence to the intrinsic error floor.

Remark. Within the emerging field of quantum operator learning, QDeepONet (Xiao et al., 2024a) represents a notable hybrid benchmark. While sharing the objective of operator approximation, our work diverges fundamentally from QDeepONet in design philosophy and theoretical grounding (see Appendix B.3 for a detailed comparison). First, QDeepONet is a hybrid architecture restricting quantum processing to linear encoding, whereas QuanONet is a pure quantum framework. This native design enables our deployment on real NISQ devices (Sec. 5.5), whereas Xiao et al. (2024a) is confined to simulation. Second, our architecture is grounded in the proposed Implicit Quadratic Frame, offering guaranteed expressivity advantages. In contrast, QDeepONet focuses primarily on computational speedup without comparable theoretical guarantees on representational capacity.

2 Related Work

2.1 Neural Operators

There has been intensive research on neural operators in recent years, motivated by the goal of learning solution operators of parametric PDE families as mappings between infinite-dimensional function spaces. In contrast to conventional surrogate models that are tied to a fixed discretization, neural operators are designed to encode input functions (e.g., coefficients, forcing terms, or boundary/initial conditions) from scattered sensor observations and to evaluate outputs at arbitrary query points, thereby supporting resolution-robust inference once training is completed. From a modeling perspective, existing methods can be broadly grouped by the structural prior used to represent the operator, including branch–trunk separable expansions, graph-based message passing constructions on unstructured discretizations, and spectral/integral-kernel parameterizations.

A representative paradigm is the Deep Operator Network (DeepONet) introduced by Lu et al. (2021), which operationalizes the universal approximation of nonlinear operators through a branch-trunk factorization. The branch net encodes the discretized input function while the trunk net encodes output coordinates, and their latent interaction yields an explicit separable representation whose effective rank is controlled by the latent width. Beyond empirical performance, the approximation and generalization behaviors of DeepONet have been studied from an infinite-dimensional viewpoint, leading to quantitative error decompositions and estimates under measure and regularity-dependent assumptions (Lanthaler et al., 2022). To reduce the dependence on paired input–output data and to impose physical consistency, physics-informed variants incorporate PDE residual and boundary/initial conditions during training, enabling operator learning even with weak or indirect supervision (Wang et al., 2021).

Another important family of approaches leverages graph-structured representations to accommodate irregular meshes, complex geometries, and heterogeneous discretizations. Graph Element Networks (Alet et al., 2019) propose to place (and potentially adapt) computational nodes in continuous domains and to propagate information via iterative message

passing, drawing an analogy to finite-element-style computation with learned connectivity and memory. Subsequent work has combined graph neural networks with differentiable PDE solvers to embed stronger inductive biases and improve predictive stability in fluid flow settings (De Avila Belbute-Peres et al., 2020). To address the prohibitive cost of capturing long-range interactions on dense graphs while preserving generalization under mesh refinement, the multipole graph neural operator introduces multilevel inducing structures inspired by fast multipole methods, achieving linear-time communication across scales and learning discretization-invariant solution operators (Li et al., 2020). More recently, message passing neural PDE solvers have been proposed as end-to-end learned computation graphs that aim to subsume classical spatial discretization schemes and to generalize across resolution, topology, and boundary conditions (Brandstetter et al., 2022).

Complementary to these spatially grounded constructions, spectral neural operators model global interactions efficiently in the frequency domain. The Fourier Neural Operator (FNO) parameterizes an integral operator by learning spectral multipliers in Fourier space, followed by pointwise nonlinearities in the physical domain (Li et al., 2021). By truncating to a fixed number of Fourier modes, FNO yields a discretization-agnostic architecture with favorable computational scaling via FFT-based implementations, and has become a standard baseline for learning parametric PDE solution operators. Collectively, these classical neural operator families can be viewed as constructing finite-dimensional latent representations based on an explicit linear frame, whose effective size scales linearly with architectural width or the number of retained modes, with predictions formed through linear combinations over the learned channels at evaluation time. This structural viewpoint provides a natural context for our study of quantum neural operators, where quantum measurements induce richer implicit quadratic feature interactions and enable fundamentally different frame constructions.

2.2 Quantum Computing for Differential Equations and Neural Operators

Quantum computing has shown great promise in solving complex problems, ranging from combinatorial optimization to quantum chemistry. In the specific domain of differential equations, emerging quantum methods have been broadly classified into two categories. The first encompasses algorithms leveraging amplitude amplification and phase estimation. For instance, methods have been developed to convert linear differential equations with non-unitary dynamics into Schrödinger-type equations, solving them via quantum phase estimation (Jin et al., 2024), or utilizing Chebyshev points for robust numerical integration (Oz et al., 2023). The second category relies on Variational Quantum Algorithms (VQA). Approaches in this paradigm optimize a cost function—typically the difference between the quantum state amplitudes and the classical solution—to approximate the evolution operator (Sarma et al., 2024; Lubasch et al., 2020). Furthermore, architectures similar to Physics-Informed Neural Networks (PINNs) have been adapted for quantum circuits, where gradients are computed through circuit evaluations to optimize parameters for specific equation instances (Kyriienko et al., 2021; Xiao et al., 2024b).

Despite these successes, current quantum solvers are predominantly restricted to individual equation instances, lacking the capability to generalize across varying parameters. The extension of quantum computing to neural operators—which learn mappings between

infinite-dimensional function spaces directly from data—remains a nascent yet critical frontier. Recently, two primary quantum neural operator architectures have been proposed: QDeepONet and Quantum Fourier Neural Operator (QFNO).

Xiao et al. (2024a) proposed the QDeepONet, which replaces the classical Branch and Trunk Nets with Quantum Orthogonal Neural Networks (QOrthoNN). By employing unary encoding and orthogonal quantum layers, this architecture theoretically reduces the evaluation complexity from quadratic $\mathcal{O}(p^2)$ to linear $\mathcal{O}(p)$ with respect to the network width (latent dimension) p. However, this computational speedup necessitates a significant trade-off in spatial resources. Specifically, the unary encoding strategy requires the number of qubits to scale linearly with the matrix dimensions, demanding approximately $1 + p$ qubits per layer. This linear scaling imposes a severe bottleneck on scalability, as deploying wide neural networks (large p) exceeds the qubit capacity of current quantum devices. Consequently, while efficient for forward propagation, the QDeepONet is currently limited to ideal simulations. In contrast, our proposed QuanONet achieves logarithmic qubit complexity $\mathcal{O}(\log p)$ and constructs a richer implicit quadratic frame. A detailed structural and functional comparison between the two architectures is provided in Appendix B.3.

Parallel to this, Jain et al. (2024) introduced the Quantum Fourier Neural Operator (QFNO). This method utilizes a sequential quantum circuit to perform a Quantum Fourier Transform (QFT) on inputs encoded as superpositions of unary states. QFNO achieves a theoretical time complexity logarithmic in the sampling dimension m, specifically $\mathcal{O}(p \log m)$, offering an exponential speedup over classical linear complexity. However, this algorithmic advantage comes at a prohibitive hardware cost. The implementation requires $p + m$ qubits and substantial circuit depth to maintain coherence, rendering it susceptible to noise and impractical for existing NISQ hardware.

2.3 Quantum Advantages and Critical Gaps in Machine Learning

The intersection of quantum mechanics and data science has catalyzed the emergence of QML, a field dedicated to exploiting phenomena such as superposition and entanglement for computational advantage (Dunjko and Briegel, 2018). Contemporary research systematically evaluates QML models through the lenses of complexity, expressivity, and trainability, identifying specific regimes where quantum approaches theoretically supersede classical counterparts.In terms of computational complexity, quantum algorithms demonstrate exponential speedups in specific linear algebra subroutines (Huang et al., 2021; Zhang et al., 2020). However, practical utility is often bottlenecked by state preparation overheads, where the cost of encoding classical data can negate algorithmic gains. Furthermore, a significant methodological risk exists in attributing quantum advantage solely to the exponential dimensionality of the Hilbert space. Without rigorous benchmarking against classical baselines operating within a comparable latent dimension p, purported speedups may prove illusory. Regarding expressivity, parameterized quantum circuits (PQCs) are rigorously characterized as partial Fourier series generators (Schuld et al., 2021). Existing literature predominantly focuses on their capacity to approximate complex continuous functions, often revealing superior parameter efficiency compared to classical neural networks in specific tasks (Yu et al., 2024). Similarly, advances in generalization suggest that quantum models may achieve lower sample complexity under certain data distributions (Caro et al.,

2022), provided that the optimization landscapes—often plagued by barren plateaus—can be effectively navigated (Du et al., 2021). However, it is worth noting that specific structural constraints in quantum circuits, such as strict unitarity, may offer inherent regularization benefits against the optimization instability often observed in high-dimensional classical networks. Despite these advancements, a critical examination of the current landscape reveals a fundamental limitation in how Quantum Neural Networks (QNNs) are conceptualized. The prevailing paradigm predominantly treats the quantum circuit as a statistical generator for scalar values. In this view, the approximation power is derived exclusively from the expectation values of measurements (i.e., $f(x) = \langle\psi(x)|H|\psi(x)\rangle$), effectively reducing the quantum system to a tool for approximating scalar functions. While successful for finite-dimensional tasks, this reductionist perspective overlooks the algebraic richness of the quantum evolution itself, failing to exploit the unitary operators as direct mathematical objects. This work addresses this critical gap. By shifting the analytical lens from the direct approximation of scalar functions to the construction of structural representations via continuous unitary matrices, we aim to unlock new capabilities for QNNs in scientific computing. This paradigm shift not only elucidates the limitations of current measurement-centric QNN applications but also serves as the heuristic foundation for the implicit quadratic frame proposed in this paper, which leverages the density matrix to span a richer feature space than standard vector-based approaches.

3 Preliminaries

3.1 The Deep Operator Network (DeepONet)

DeepONet (Lu et al., 2019) is a deep learning framework engineered to approximate nonlinear operators. Its architecture is theoretically grounded in the universal approximation theorem for operators (Chen and Chen, 1995).

The DeepONet consists of two sub-architectures: the *Branch (BNC) Net*, encoding the input function u (discretized as $\tilde{\mathbf{u}} = [u(\mathbf{x}_1), \ldots, u(\mathbf{x}_m)]^\top \in \mathbb{R}^m$), and the *Trunk (TNK) Net*, encoding the output coordinate $\mathbf{y} \in \mathbb{R}^{d_2}$. Let $\mathcal{N}$ denote a generic deep neural network parameterized by trainable weights $\boldsymbol{\theta}$. The outputs of the Branch and Trunk nets, $\mathbf{b}(u), \mathbf{t}(\mathbf{y}) \in \mathbb{R}^p$, are defined as:

$$\begin{aligned} \mathbf{b}(u) &= \mathcal{N}_{\text{BNC}}(\tilde{\mathbf{u}}; \boldsymbol{\theta}_b), \\ \mathbf{t}(\mathbf{y}) &= \mathcal{N}_{\text{TNK}}(\mathbf{y}; \boldsymbol{\theta}_t). \end{aligned} \tag{1}$$

In the original formulation (Lu et al., 2021), these components are instantiated as Multilayer Perceptrons (MLPs). The final operator approximation is obtained by the dot product:

$$\mathscr{D}(u)(\mathbf{y}) = \mathbf{b}(u) \cdot \mathbf{t}(\mathbf{y}) + b_0 \approx \mathscr{G}(u)(\mathbf{y}). \tag{2}$$

This architecture is supported by the universal approximation theorem:

Theorem 1 (Universal Approximation of DeepONet (Chen and Chen, 1995)) *Let K_1 be a compact subset of a Banach space of input functions, $D_2 \subset \mathbb{R}^{d_2}$ be a compact domain, and $\mathscr{G} : K_1 \to \mathrm{C}(D_2)$ be a continuous nonlinear operator. For any $\epsilon > 0$, there exists*

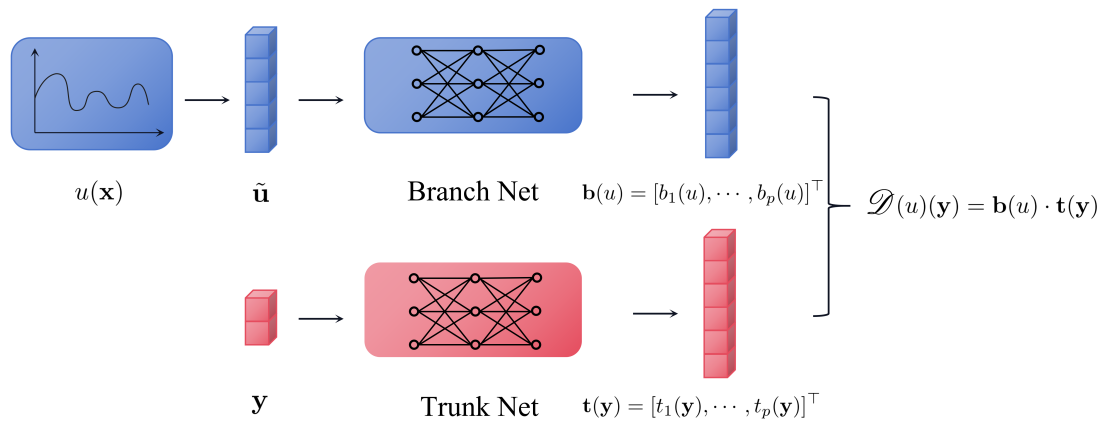

Figure 1: Architecture of DeepONet, consisting of a Branch Net for encoding input functions and a Trunk Net for coordinate encoding.

a Deep Operator Network $\mathscr{D}$ parameterized by a latent dimension p such that:

$$\sup_{u \in K_1, \mathbf{y} \in D_2} \left| \mathscr{G}(u)(\mathbf{y}) - \sum_{k=1}^{p} b_k(u) t_k(\mathbf{y}) \right| < \epsilon. \tag{3}$$

DeepONet exhibits strong generalization capabilities, allowing for rapid inference on unseen initial or boundary conditions without retraining. Extensions such as the Unstacked DeepONet (Lu et al., 2021), which utilizes a shared Branch Net to reduce computational costs, and the Sequential DeepONet (S-DeepONet) (He et al., 2024), which incorporates sequential learning for time-dependent problems, have further broadened its applicability.

3.2 Theoretical Framework for Operator Learning Error

Following the unified framework proposed in Lanthaler et al. (2022), the DeepONet architecture can be formalized through three distinct mappings:

- **Encoder ($\mathscr{E}$).** A linear mapping that discretizes the input function u at sensor locations:
$$\mathscr{E} : \mathrm{C}(D_1) \to \mathbb{R}^m, \quad \mathscr{E}(u) = \tilde{\mathbf{u}} = [u(\mathbf{x}_1), \ldots, u(\mathbf{x}_m)]. \tag{4}$$

- **Approximator ($\mathscr{A}$).** A neural network mapping the discretized input to the latent feature space $\mathbb{R}^p$:
$$\mathscr{A} : \mathbb{R}^m \to \mathbb{R}^p, \quad \mathscr{A}(\mathscr{E}(u)) = \mathbf{b}(u). \tag{5}$$
The composition $\mathscr{B}(u) = \mathscr{A} \circ \mathscr{E}(u)$ constitutes the Branch Net.

- **Reconstructor ($\mathscr{R}$).** Utilizing the Trunk Net $\mathscr{T} : D_2 \to \mathbb{R}^p$, the reconstructor maps the latent features back to the function space $\mathrm{C}(D_2)$:
$$\mathscr{R} : \mathbb{R}^p \to \mathrm{C}(D_2), \quad \mathscr{R}(\mathbf{b})(\mathbf{y}) := \mathbf{b} \cdot \mathscr{T}(\mathbf{y}) = \sum_{k=1}^{p} b_k t_k(\mathbf{y}). \tag{6}$$

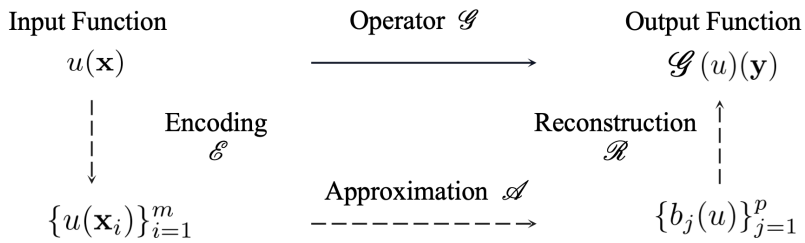

Figure 2: Schematic illustration of operator learning via the Encoder-Approximator-Reconstructor framework.

The complete DeepONet operator $\mathscr{D}$ is thus the composition:

$$\mathscr{D} : \mathrm{C}(D_1) \to \mathrm{C}(D_2), \quad \mathscr{D}(u) = (\mathscr{R} \circ \mathscr{A} \circ \mathscr{E})(u). \tag{7}$$

Crucially, theoretical analysis proves that the reconstruction error is strictly bounded by the spectral decay of the output operator. Specifically, Theorem 3.4 in Lanthaler et al. (2022) establishes a sharp lower bound:

$$\hat{\mathcal{E}}_{\mathscr{R}} \geq \sqrt{\sum_{k>p} \lambda_k}, \tag{8}$$

where $\lambda_1 \geq \lambda_2 \geq \dots$ are the eigenvalues of the covariance operator associated with the push-forward measure $\mathscr{G}_{\#}\mu$. Furthermore, Theorem 3.10 therein confirms that the optimal Trunk Net basis functions correspond exactly to the leading p eigenfunctions (Principal Components) of the output space.

This implies that DeepONet is theoretically confined to an explicit linear frame of rank p. To reduce the error floor defined by the tail sum $\sum_{k>p} \lambda_k$, one is mathematically forced to linearly increase the latent dimension p. For complex physical systems with slow spectral decay (e.g., advection-dominated flows or high-frequency oscillations), this linear scaling becomes prohibitively expensive. This limitation motivates our proposed quantum architecture, which constructs an implicit frame of effective rank $\mathcal{O}(p^2)$ without increasing the physical latent dimension p.

4 Methodology

In this section, we derive the quantum approximation theorems that underpin our framework. We begin by extending the classical operator approximation theory to the domain of quantum states. Subsequently, we formulate the quantum universal approximation theorem for operators using the Frobenius inner product of density matrix, which aligns with physical quantum measurement protocols. Crucially, we analyze the impact of physical constraints on the expressivity of the quantum form. We further demonstrate that the Trainable-Frequency strategy is a structural necessity to prevent dimensional collapse, thereby ensuring the implicit frame spans the $\mathcal{O}(p^2)$ dimension.

4.1 Theoretical Results

We now establish the theoretical foundation of QuanONet. Our derivation begins with the fundamental observation—central to our preliminary study (Wang et al., 2025)—that the inner product structure required by classical operator theories can be geometrically embedded into the Hilbert space of quantum states.

Theorem 2 (Embedding of Vector Inner Products into Quantum States) *Let $D_1 \subset \mathbb{R}^{d_1}$ and $D_2 \subset \mathbb{R}^{d_2}$ be compact domains. Let $K_1 \subset \mathrm{C}(D_1)$ be a compact subset of the continuous function space. Suppose that $\mathbf{b} : K_1 \to \mathbb{R}^p$ is a continuous vector-valued functional and $\mathbf{t} : D_2 \to \mathbb{R}^p$ is a continuous vector-valued function.*

Provided that $\mathbf{b}$ and $\mathbf{t}$ have non-trivial norms, there exists a scaling constant $\lambda \in \mathbb{R}$ and normalized vectors $|b(u)\rangle, |t(\mathbf{y})\rangle \in \mathbb{C}^{p+2}$ such that for all $u \in K_1$ and $\mathbf{y} \in D_2$:

$$\mathbf{b}(u) \cdot \mathbf{t}(\mathbf{y}) = \lambda \langle b(u)|t(\mathbf{y})\rangle. \tag{9}$$

Theorem 2 confirms that the quantum feature space inherently possesses the capacity to represent the dot-product operations essential for neural operators. While this geometric correspondence served as the theoretical basis for the architecture proposed in Wang et al. (2025), transforming it into a physically efficient protocol presents challenges. A direct implementation via the SWAP test (Buhrman et al., 2001) imposes significant hardware overhead, necessitating two distinct quantum registers and ancillary qubits (scaling as $2 \log p$). Furthermore, standard projective measurements yield transition probabilities proportional to the squared modulus $|\langle b(u)|t(\mathbf{y})\rangle|^2$, rather than the linear inner product required by the summation form $\sum_k b_k(u) t_k(\mathbf{y})$.

To overcome these physical constraints and eliminate the need for complex subroutines like Quantum Phase Estimation (QPE), we advance beyond the state overlap picture utilized in Wang et al. (2025). In this work, we rigorously formulate the approximation theory based on the expectation value (Trace) formalism. We demonstrate that the trace of the product of a modulated observable and a density matrix not only recovers the required linear structure but does so with superior hardware efficiency.

Theorem 3 (Universal Approximation via Quantum Measurement) *Let K_1 be a compact subset of a Banach space of input functions, and $D_2 \subset \mathbb{R}^{d_2}$ be a compact domain of output coordinates. Let $\mathscr{G} : K_1 \to \mathrm{C}(D_2)$ be a continuous nonlinear operator. For any $\epsilon > 0$, there exist:*

1. *An observable $\mathbf{M}(u) = \mathbf{B}(u)\mathbf{\Lambda}\mathbf{B}^\dagger(u)$ which encodes the input function u with the fixed spectrum $\mathbf{\Lambda}$, and*

2. *A rank-1 density matrix $\mathbf{D}(\mathbf{y}) = \mathbf{t}(\mathbf{y})\mathbf{t}^\dagger(\mathbf{y})$ which encodes the output coordinate $\mathbf{y}$, derived from a pure quantum state $|t(\mathbf{y})\rangle \in \mathcal{H}$,*

such that the operator $\mathscr{G}$ can be approximated by the Frobenius inner product of their product:

$$\sup_{u \in K_1, \mathbf{y} \in D_2} |\mathscr{G}(u)(\mathbf{y}) - \langle \mathbf{M}(u), \mathbf{D}(\mathbf{y})\rangle_F| < \epsilon. \tag{10}$$

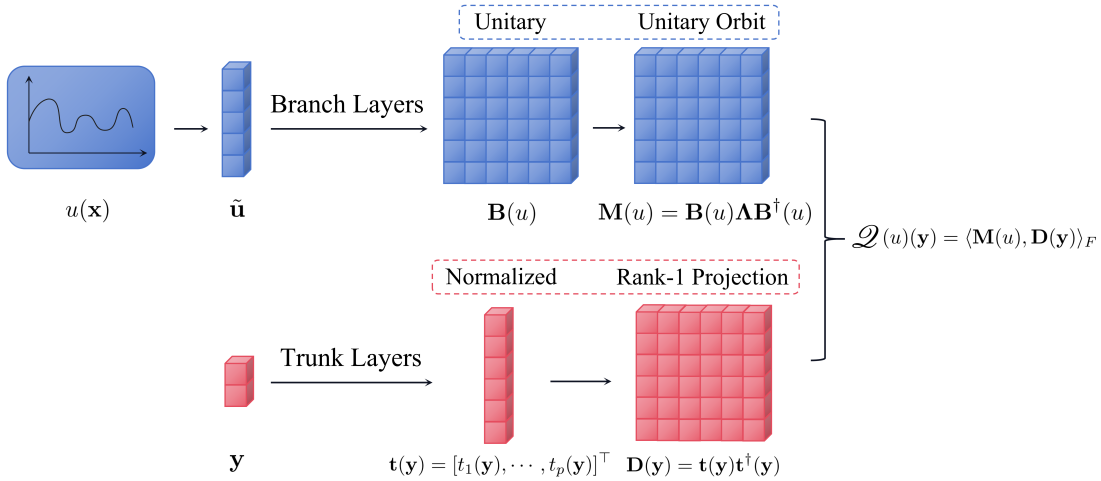

Figure 3: Architecture of QuanONet. The Branch layers encode the input function $u(\mathbf{x})$ into a Branch-Modulated Observable $\mathbf{M}(u)$, while the Trunk layers map the coordinate $\mathbf{y}$ to a Trunk-Density Matrix $\mathbf{D}(\mathbf{y})$ constructed from the output quantum state. The final operator value is computed via the Frobenius inner product. The constraints in the dashed box are from the quantum mechanisms.

The proof leverages the isometric isomorphism between the space of linear operators $\mathcal{L}(\mathcal{H})$ (equipped with the Frobenius inner product $\langle \mathbf{A}, \mathbf{B}\rangle_F := \mathrm{Tr}[\mathbf{A}^\dagger \mathbf{B}]$) and a high-dimensional Euclidean space. Since the classical universal approximation theorem guarantees approximation via finite dot products in Euclidean space, we can embed this structure directly into the quantum operator space. Specifically, in QuanONet, $\mathbf{M}(u)$ is Hermitian and $\mathbf{D}(\mathbf{y})$ is a rank-1 projector, both subject to constraints from quantum mechanisms. Detailed derivations regarding the embedding and the sufficiency of the physically constrained operators are provided in Appendix C.3.

Theorem 3 constitutes the primary theoretical justification for QuanONet. It demonstrates that the model's output $\mathscr{Q}(u)(\mathbf{y}) = \langle \mathbf{M}(u), \mathbf{D}(\mathbf{y})\rangle_F$ is a mathematically rigorous approximator. Physically, this operation corresponds to the expectation value measurement of the observable $M(u)$ with respect to the quantum state described by the density matrix $\mathbf{D}(\mathbf{y})$, thereby bridging the abstract operator approximation theory with concrete quantum measurement protocols. Specifically, as derived in Appendix C.3, a quantum system of dimension p provides a sufficient representational capacity to approximate target operators requiring up to $p-2$ independent functions, thereby guaranteeing a representational power of at least order $\mathcal{O}(p)$ with respect to the Hilbert space dimension.

4.2 The Architecture of QuanONet

Consequently, we propose QuanONet based on this paradigm. By sequentially embedding the input data $\tilde{\mathbf{u}} = [u(\mathbf{x}_1), \ldots, u(\mathbf{x}_m)]^\top$ and output coordinates $\mathbf{y}$ into a single quantum register, QuanONet evaluates the target operator using only $\log p$ qubits, thereby significantly reducing circuit width while preserving approximation guarantees.

As illustrated in Fig. 4, QuanONet consists of two primary modules: the *Branch (BNC) layers* and the *Trunk (TNK) layers*, which respectively encode the input function $\tilde{\mathbf{u}}$ and the output coordinates $\mathbf{y}$. The depth of these networks is governed by the hyperparameters h_b and h_t, which denote the number of repetitions of the embedding layers, while l_b and l_t represent the ansatz depth for the Branch and Trunk layers, respectively. The circuit output is given by

$$\begin{aligned}\mathscr{Q}(u)(\mathbf{y}) =& \langle \psi_{\boldsymbol{\theta}}(\mathbf{y}, \tilde{\mathbf{u}}) | H | \psi_{\boldsymbol{\theta}}(\mathbf{y}, \tilde{\mathbf{u}}) \rangle \\ =& \langle 0 | U_{\mathrm{TNK}}^{\dagger}(\boldsymbol{\theta}^t, \mathbf{y}) U_{\mathrm{BNC}}^{\dagger}(\boldsymbol{\theta}^b, \tilde{\mathbf{u}}) H \, U_{\mathrm{BNC}}(\boldsymbol{\theta}^b, \tilde{\mathbf{u}}) U_{\mathrm{TNK}}(\boldsymbol{\theta}^t, \mathbf{y}) | 0 \rangle.\end{aligned} \tag{11}$$

Expressing the Hamiltonian in its spectral decomposition, $H = \sum_{i=1}^{p} \lambda_i |\phi_i\rangle\langle\phi_i|$, the output becomes

$$\mathscr{Q}(u)(\mathbf{y}) = \sum_{i=1}^{p} \lambda_i \left| \langle \phi_i | U_{\mathrm{BNC}}(\boldsymbol{\theta}^b, \tilde{\mathbf{u}}) U_{\mathrm{TNK}}(\boldsymbol{\theta}^t, \mathbf{y}) | 0 \rangle \right|^2. \tag{12}$$

We define the Branch and Trunk states as

$$|b_i(u)\rangle = U_{\mathrm{BNC}}^{\dagger}(\tilde{\mathbf{u}}) |\phi_i\rangle = \sum_{j=1}^{p} B_{ij}(u) |j\rangle, \tag{13}$$

$$|t(\mathbf{y})\rangle = U_{\mathrm{TNK}}(\mathbf{y}) |0\rangle = \sum_{j=1}^{p} t_j(\mathbf{y}) |j\rangle. \tag{14}$$

Thus, the output can be written as

$$\mathscr{Q}(u)(\mathbf{y}) = \langle t(\mathbf{y}) | \left(\sum_{i=1}^{p} \lambda_i |b_i(u)\rangle\langle b_i(u)| \right) |t(\mathbf{y})\rangle \tag{15}$$

$$= \sum_{i=1}^{p} \lambda_i |\langle b_i(u) | t(\mathbf{y})\rangle|^2. \tag{16}$$

The QuanONet output $\mathscr{Q}(u)(\mathbf{y})$ is thus derived from the quantum circuit that computes the expectation value of a parameterized observable with respect to a parameterized quantum state (Wang et al., 2025).

Let $H = \sum_{i=1}^{p} \lambda_i |\phi_i\rangle\langle\phi_i|$ denote the spectral decomposition of the Hamiltonian, where $\{|\phi_i\rangle\}_{i=1}^{p}$ forms an orthonormal basis of the Hilbert space $\mathcal{H} \simeq \mathbb{C}^p$. Here, the Hilbert space dimension corresponds to the latent dimension p of the operator approximation. We use this basis consistently to represent all vectors and matrices.

Under this convention, each quantum state $|x\rangle \in \mathcal{H}$ is represented by its coordinate vector

$$\mathbf{x} = [\langle\phi_1|x\rangle, \langle\phi_2|x\rangle, \ldots, \langle\phi_p|x\rangle]^{\top}, \tag{17}$$

and any linear operator $A : \mathcal{H} \to \mathcal{H}$ is represented $\mathbf{A}$ by the matrix with elements

$$A_{ij} = \langle\phi_i|A|\phi_j\rangle, \qquad 1 \le i, j \le p. \tag{18}$$

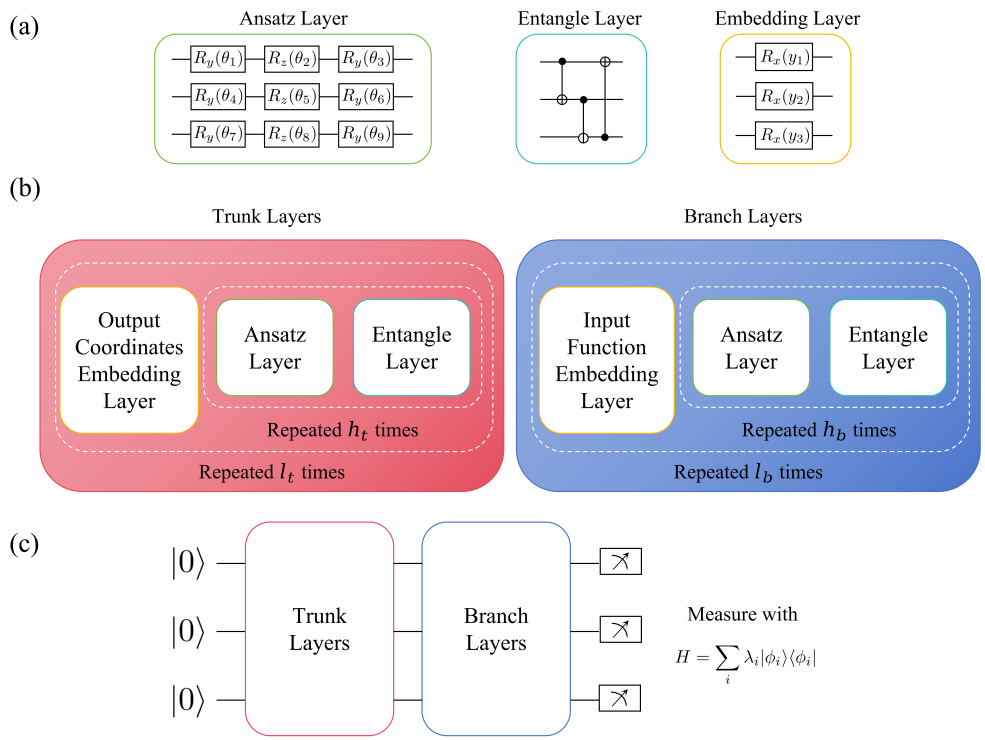

Figure 4: QuanONet's detailed architecture. (a) A hardware-efficient class of examples of Embedding, Entangle, and Ansatz layers. (b) Structure of Branch and Trunk layers with hyperparameters h_b, h_t, l_b, l_t representing the number of repetitions of Ansatz-Entangle and Embedding-Train alternating layers in them. (c) The overall quantum circuit structure. The circuit parameters are trained by gradient descent.

In this basis set, the Hamiltonian H becomes a $p \times p$ diagonal matrix

$$\mathbf{\Lambda} = \mathrm{diag}[\lambda_1, \ldots, \lambda_p]. \tag{19}$$

All matrix and vector representations appearing in this work are taken with respect to this basis set $\{|\phi_i\rangle\}_{i=1}^p$. For simplicity, we identify $|x\rangle$ with its column vector $\mathbf{x}$, and linear operators with their corresponding matrices.

We now define the two core matrix components of QuanONet.

First, the *Trunk-Density Matrix* $\mathbf{D}(\mathbf{y})$ is defined as the $p \times p$ density matrix associated with the pure Trunk state $|t(\mathbf{y})\rangle$. The elements of $\mathbf{D}(\mathbf{y})$ are given by $D_{jk}(\mathbf{y}) = t_j(\mathbf{y})t_k^*(\mathbf{y})$, i.e.,

$$\mathbf{D}(\mathbf{y}) = \mathbf{t}(\mathbf{y})\mathbf{t}^\dagger(\mathbf{y}), \tag{20}$$

where $\mathbf{t}(\mathbf{y}) = [t_1(\mathbf{y}), \ldots, t_p(\mathbf{y})]^\top$ is the column vector of Trunk-state amplitudes.

Second, we define the *Branch-Modulated Observable* $\mathbf{M}(u)$ through its matrix representation. Let $\mathbf{B}(u)$ denote the $p \times p$ unitary matrix that transforms $\{|\phi_i\rangle\}_{i=1}^p$ into $\{|b_i(u)\rangle\}_{i=1}^p$:

$$\mathbf{B}(u) = \begin{bmatrix} | & & | \\ \mathbf{b}_1(u) & \cdots & \mathbf{b}_p(u) \\ | & & | \end{bmatrix}, \tag{21}$$

which is the matrix form of $U_{\text{BNC}}^{\dagger}$. The modulated observable then takes the compact matrix form

$$\mathbf{M}(u) = \mathbf{B}(u)\mathbf{\Lambda}\mathbf{B}^{\dagger}(u). \tag{22}$$

Substituting these definitions, the final output of QuanONet can be expressed in the trace form

$$\begin{aligned}\mathscr{Q}(u)(\mathbf{y}) &= \sum_{i=1}^{p} \lambda_i |\mathbf{b}_i(u) \cdot \mathbf{t}(\mathbf{y})|^2 = \langle \mathbf{M}(u), \mathbf{D}(\mathbf{y}) \rangle_F \\ &= \mathrm{Tr}[\mathbf{M}(u)\mathbf{D}(\mathbf{y})] = \sum_{j,k=1}^{p} M_{jk}^{*}(u) D_{jk}(\mathbf{y}),\end{aligned} \tag{23}$$

which corresponds to the expectation value of $|t(\mathbf{y})\rangle$ on the modulated observable. The final step follows from the fact that $\mathbf{M}$ is Hermitian. Similar to DeepONet, QuanONet also adds a trainable bias b_0 to the output.

$$\mathscr{Q}(u)(\mathbf{y}) = \langle \mathbf{M}(u), \mathbf{D}(\mathbf{y}) \rangle_F + b_0 \approx \mathscr{G}(u)(\mathbf{y}). \tag{24}$$

As a general framework, QuanONet admits various hardware-specific implementations. To address current hardware limitations—such as restricted qubit connectivity and limited circuit depth—we employ a *Hardware-Efficient Ansatz* (HEA) (Kandala et al., 2017), as illustrated in Fig. 4. This design achieves a practical balance between expressivity and resource efficiency, enabling implementation on near-term quantum devices.

4.3 Comparative Analysis of Model Expressivity and Constraints

QuanONet follows the general structure of DeepONet proposed in Lanthaler et al. (2022) and comprises three distinct components:

- **Encoder ($\mathscr{E}$):** Identical to the first step of DeepONet, the encoder maps the input function u evaluated at m sensor locations to a finite-dimensional feature vector:

$$\mathscr{E} : \mathrm{C}(D_1) \to \mathbb{R}^m, \quad \mathscr{E}(u) = \tilde{\mathbf{u}} = [u(\mathbf{x}_1), \ldots, u(\mathbf{x}_m)]. \tag{25}$$

- **Approximator ($\mathscr{A}_Q$):** Given the encoded input, the Branch layers of the QNN learn a mapping to the unitary orbit $\mathcal{M}_{\mathbf{\Lambda}}$ of the fixed spectrum $\mathbf{\Lambda}$:

$$\mathscr{A}_Q : \mathbb{R}^m \to \mathcal{M}_{\mathbf{\Lambda}} \subset \mathcal{L}(\mathcal{H}), \quad \mathscr{A}_Q(\mathscr{E}(u)) = \mathbf{M}(u). \tag{26}$$

  Physically, this mapping is parameterized by a unitary matrix $\mathbf{B}(u) \in \mathrm{SU}(p)$ generated by the quantum circuit, constructing the *Branch-Modulated Observable* via spectral conjugation:

$$\mathbf{M}(u) = \mathbf{B}(u)\mathbf{\Lambda}\mathbf{B}^{\dagger}(u). \tag{27}$$

- **Reconstructor ($\mathscr{R}_Q$):** The Trunk layers define a mapping to the complex projective space via a rank-1 projection:

$$\mathscr{T}_Q : \mathbb{R}^{d_2} \to \mathbb{C}P^{p-1}, \quad \mathscr{T}_Q(\mathbf{y}) = \mathbf{D}(\mathbf{y}) = \mathbf{t}(\mathbf{y})\mathbf{t}^{\dagger}(\mathbf{y}), \tag{28}$$

for any coordinate $\mathbf{y} \in D_2 \subset \mathbb{R}^{d_2}$, effectively removing the global phase ambiguity of the state vector. The final operator reconstruction is defined as the Frobenius inner product between the observable $\mathbf{M}(u)$ and the projective representation $\mathbf{D}(\mathbf{y})$:

$$\mathscr{R}_Q : \mathcal{M}_{\mathbf{\Lambda}} \to \mathrm{C}(D_2), \quad \mathscr{R}_Q(\mathbf{M}) := \langle \mathbf{M}, \mathbf{D}(\mathbf{y}) \rangle_F = \sum_{j,k=1}^{p} M_{jk}^* D_{jk}(\mathbf{y}). \tag{29}$$

Combining these components yields the overall QuanONet operator mapping:

$$\mathscr{Q} : \mathrm{C}(D_1) \to \mathrm{C}(D_2), \quad \mathscr{Q}(u) = (\mathscr{R}_Q \circ \mathscr{A}_Q \circ \mathscr{E})(u). \tag{30}$$

A central claim of this work is that the proposed architecture possesses a significantly richer expressive capacity than the classical DeepONet framework (Lanthaler et al., 2022), despite both being parameterized by a latent dimension p. This superior expressivity stems from the quadratic nature of the QuanONet kernel, which implicitly utilizes $\mathcal{O}(p^2)$ feature functions and coefficients, in contrast to the linear $\mathcal{O}(p)$ structure of DeepONet. To substantiate this claim, we analyze the geometric constraints imposed on the two core matrices:

- **Trunk-Density Matrix ($\mathbf{D}(\mathbf{y})$):** The matrix $\mathbf{D}(\mathbf{y}) = \mathbf{t}(\mathbf{y})\mathbf{t}^\dagger(\mathbf{y})$ is constrained to be a rank-1 projection matrix. The manifold of rank-1 projectors is isomorphic to the complex projective space $\mathbb{C}P^{p-1}$.
- **Branch-Modulated Observable ($\mathbf{M}(u)$):** The matrix $\mathbf{M}(u)$ is constrained to the unitary orbit $\mathcal{M}_{\mathbf{\Lambda}}$ generated by a fixed diagonal spectrum $\mathbf{\Lambda}$.

Regarding $\mathbf{D}(\mathbf{y})$, although it is algebraically constrained to rank-1, the trace contraction $\mathrm{Tr}[\mathbf{M}(u)\mathbf{D}(\mathbf{y})]$ implies that the effective frame functions are the pairwise products $\{t_i(\mathbf{y})t_j^*(\mathbf{y})\}_{i,j}$. This constitutes a quadratic extension of the primitive frame. While the constructive proof in Appendix C.3 utilizes a specialized sparse structure to guarantee a baseline linear capacity of $\mathcal{O}(p)$, this underlying quadratic mechanism implicitly constructs a frame of size $\mathcal{O}(p^2)$, suggesting that the model's empirical expressivity significantly exceeds the theoretical lower bound.

Regarding $\mathbf{M}(u)$, the effective degrees of freedom available for coefficient approximation are determined by the dimension of the unitary orbit. Let $\mathbf{\Lambda}$ have distinct eigenvalues with multiplicities $n_1, \ldots, n_q$. The orbit dimension is given by:

$$\dim(\mathcal{M}_{\mathbf{\Lambda}}) = p^2 - \sum_{j=1}^{q} n_j^2. \tag{31}$$

To volumetrically maximize the feasible coefficient region and avoid topological obstructions caused by dimensional collapse, it is imperative to initialize the system with a strictly non-degenerate spectrum (where $n_j = 1$ for all j). This maximizes the orbit dimension to $p^2 - p$, thereby providing sufficient freedom to fully utilize the capacity of the implicit quadratic frame.

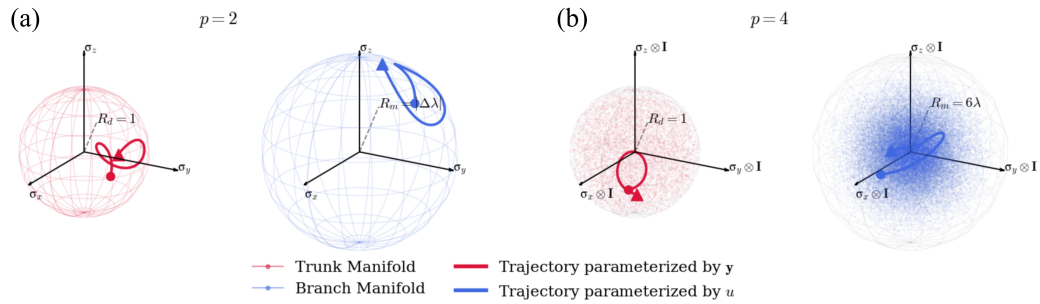

Figure 5: Visualization of the manifolds $\mathbf{D}$ and $\mathbf{M}$ projected onto the first qubit Pauli basis. (a) For $p = 2$, the geometry is restricted to a spherical surface (Bloch sphere). (b) For $p = 4$, the projection manifests as a solid volumetric ball. In both regimes, a sufficiently large spectral radius is required to cover the target codomain. For the $p = 4$ case, the spectrum is set to $\mathbf{\Lambda} = \mathrm{diag}[-2\lambda, -\lambda, \lambda, 2\lambda]$, and unitary matrices are sampled from the uniform Haar measure.

To concretize the abstract constraints of the unitary orbit $\mathcal{M}_{\mathbf{\Lambda}}$, we consider the minimal case of a single-qubit system ($p = 2$). This case is particularly illustrative because both $\mathbf{M}$ and $\mathbf{D}$ effectively constitute 2-dimensional manifolds; specifically, $\mathbf{D}$ can be viewed as a special instance of $\mathbf{M}$ with a fixed spectrum of $\{1, 0\}$.

With a fixed spectrum $\mathbf{\Lambda} = \mathrm{diag}[\lambda_1, \lambda_2]$, any Hermitian matrix $\mathbf{M}$ in this space can be decomposed into the normalized Pauli basis $\{\frac{1}{2}\mathbf{I}, \frac{1}{2}\boldsymbol{\sigma}\}$ as $\mathbf{M}(u) = m_0(u)\frac{\mathbf{I}}{2} + \mathbf{m}(u) \cdot \frac{\boldsymbol{\sigma}}{2}$. The spectral constraints impose rigid conservation laws: the fixed trace determines the bias component $m_0 = \lambda_1 + \lambda_2$, while the fixed determinant constrains the coefficient vector $\mathbf{m}(u)$ to a sphere of radius $R_m = |\lambda_1 - \lambda_2| = \Delta\lambda$. Similarly, the Trunk-Density Matrix $\mathbf{D}(\mathbf{y})$, representing a pure state projector, corresponds to a vector on the Bloch sphere with radius $R_d = 1$ and a fixed bias $d_0 = 1$. Consequently, the Frobenius inner product in QuanONet decomposes into a spherical inner product plus a constant bias:

$$\langle \mathbf{M}(u), \mathbf{D}(\mathbf{y}) \rangle_F = \underbrace{m_0 d_0}_{\text{Fixed Bias}} + \underbrace{\mathbf{m}(u) \cdot \mathbf{d}(\mathbf{y})}_{\text{Bloch Sphere}}. \tag{32}$$

As shown in Fig. 5(a), the geometry in the 3-dimension space manifests as two spheres. Despite the rigid norm constraints restricting $\mathbf{m}$ and $\mathbf{d}$ to these surfaces, the system retains the capacity to approximate unconstrained target values, provided the spectral gap $\Delta\lambda$ is sufficiently large. This geometric mechanism directly parallels the embedding principle in Theorem 2, where arbitrary non-normalized inner products are represented by normalized quantum states via an appropriate scaling factor; here, a large spectral radius R_m plays a similar role, effectively expanding the feasible projection region to cover the target codomain.

Extending this geometric intuition to higher dimensions ($p = 4$), we examine the projection of the 15-dimensional manifold $\mathbf{M}$ and 6-dimensional manifold $\mathbf{D}$ onto the Pauli vectors of the first qubit. Unlike the hollow spherical surface observed in the single-qubit case, the projected geometry of the 2-qubit observable manifests as a solid volumetric ball in the reduced 3-dimensional subspace. Specifically, we consider a non-degenerate spectrum for

$\mathbf{M}$ given by $\mathbf{\Lambda} = \mathrm{diag}[-2\lambda, -\lambda, \lambda, 2\lambda], \lambda > 0$. As illustrated in Fig. 5(b), although the full manifold resides in a 15-dimensional space (SU(4)), their local projection on the first qubit $\mathbf{m} = \mathrm{Tr}[\mathbf{M}(\boldsymbol{\sigma} \otimes \mathbf{I})]$ and $\mathbf{d} = \mathrm{Tr}[\mathbf{D}(\boldsymbol{\sigma} \otimes \mathbf{I})]$ fill a ball of radius $R_m = 6\lambda$. Similar to the $p = 2$ case, a sufficiently large spectral scale λ is requisite to inflate this volumetric ball, ensuring it encompasses the target domain within the local observable space. It is noteworthy that under the Haar measure—which samples unitaries U uniformly from SU(4)—the projected manifold $\mathbf{M}$ manifests as a distribution that is significantly denser at the ball's center; in the extreme limit of spectral degeneracy, this volume would collapse entirely to a single point, thereby exhibiting a degradation of the manifold's dimension.

4.4 Trainable-Frequency Embedding Strategy

We adopt a novel perspective on the embedding strategy, diverging from the conventional view of *Trainable-Frequency* (TF) (Jaderberg et al., 2024) as merely a method to optimize the spectral alignment of linear quantum state components. Instead, we identify it as a structural necessity to enhance the spectral expressivity of the implicit density matrix, thereby unlocking the full potential of the quadratic frame. To rigorously ground this in QuanONet, we invoke the theoretical framework treating parameterized quantum circuits as partial Fourier series (Schuld et al., 2021). In this representation, the quantum model output is expressed as a linear combination of Fourier modes determined by the eigenvalues of the data-embedding Hamiltonians[2].

Standard quantum architectures typically employ a *Fixed-Frequency* (FF) strategy, where embedding weights are locked to integer multiples of a base frequency (e.g., $\Omega_{\mathbf{t}} = \{k\omega_0 \mid k \in \mathbb{Z}\}$). While sufficient for standard QNNs, we demonstrate that this strategy induces a dimensional collapse in the context of QuanONet's quadratic frame. The frame functions of the Trunk-Density Matrix, defined as $D_{ij}(\mathbf{y}) = t_i(\mathbf{y})t_j^*(\mathbf{y})$, generate frequencies in the difference set $\Omega_{\mathbf{D}} = \{\omega_i - \omega_j \mid \omega_i, \omega_j \in \Omega_{\mathbf{t}}\}$. Crucially, for a fixed integer grid, this difference set suffers from severe spectral redundancy. Instead of expanding quadratically, the number of unique frequencies is restricted to scale linearly as roughly twice the latent dimension $(2p - 1)$, as illustrated in Fig. 6(a).

To resolve this bottleneck and unlock the potential of the implicit quadratic frame, we adopt the *Trainable-Frequency* (TF) paradigm (Jaderberg et al., 2024). For a generic input vector $\mathbf{z} \in \{\tilde{\mathbf{u}}, \mathbf{y}\}$, the embedding is defined as:

$$\mathbf{z} \mapsto \boldsymbol{\omega} \odot \mathbf{z} + \boldsymbol{\kappa}, \tag{33}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{\kappa}$ are trainable weight and bias vectors. By treating the embedding frequencies $\boldsymbol{\omega}$ as continuous trainable parameters, the effective spectral modes become non-commensurate real numbers. In this regime, the pairwise spectral differences $\{\omega_i - \omega_j\}_{i,j}$ are almost surely distinct, ensuring that the resulting frame functions $\{D_{ij}\}$ span a spectral support of size $\mathcal{O}(p^2)$, as shown in Fig. 6(b). This effectively expands the spectral bandwidth of the model, maximizing the representational reach of the quadratic construction.

This spectral expansion is the pivotal condition that confers a fundamental advantage in reconstruction efficiency. Specifically, while the independent degrees of freedom remain

2. This construction invites a theoretical comparison with the fixed trigonometric basis used in FNO. See Appendix B.4 for a detailed analysis of their structural commonalities and differences.

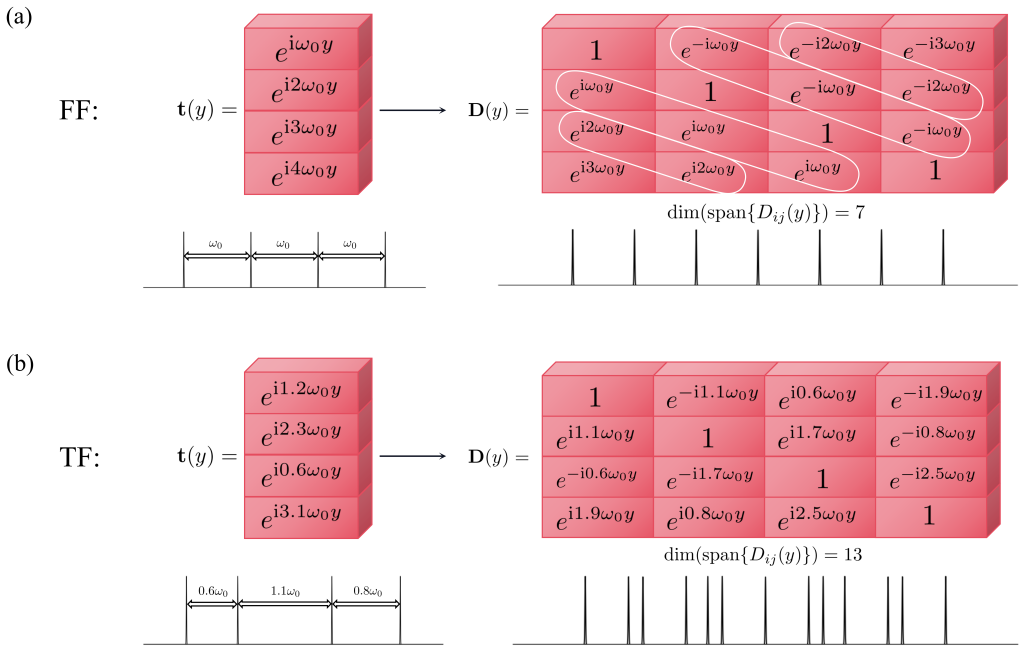

Figure 6: Illustrative visualization of Trunk layer outputs for a latent dimension $p = 4$. For clarity, the state normalization factors are omitted to focus on the spectral distribution. (a) The FF model employs harmonic functions $\{e^{ik\omega_0 y}\}_{k=1}^{4}$. The resulting density matrix $\mathbf{D}(y)$ exhibits high redundancy (marked by white circles) due to frequency overlaps, collapsing the effective frame dimension to a linear scaling (dim $= 7$). (b) The TF model utilizes non-commensurate frequencies. This breaks the spectral symmetry, yielding a dense frame with dimension 13, corresponding to the maximal difference set cardinality $p^2 - p + 1$ defined by a Sidon set.

$\mathcal{O}(p)$, the TF strategy expands the effective frequency support from $\mathcal{O}(p)$ to $\mathcal{O}(p^2)$. Consider the reconstruction error $\hat{\mathcal{E}}_{\mathscr{R}_Q}$ governed by the spectral decay of the target operator. For a classical linear reconstruction utilizing p latent dimensions, the model is strictly band-limited to the leading p eigenvalues, yielding an error typically bounded by $\mathcal{O}(p^{-s/d})$ for operators mapping to Sobolev spaces H^s. In stark contrast, because the implicit frame $\{t_i(\mathbf{y})t_j^*(\mathbf{y})\}_{i,j}$ spans a significantly larger spectral bandwidth of size $\mathcal{O}(p^2)$, the architecture acquires the capacity to capture frequency correlations that lie far beyond the reach of the classical linear basis. Although the coefficients on this frame are constrained by the rank-1 structure of the density matrix, this approximation capability allows the model to adaptively align its degrees of freedom with the dominant spectral components within this expanded support. Consequently, for operators exhibiting spectral sparsity or specific correlation structures, the convergence behavior can surpass the classical barrier, effectively approaching the efficiency of a truncation threshold significantly larger than p.

5 Experiments

5.1 Protocols

5.1.1 Hardware and Software Setup

All numerical simulations were performed on a high-performance workstation equipped with an AMD Ryzen Threadripper 3970X CPU (32 physical cores), 190 GB RAM, and five NVIDIA GeForce RTX 3090 GPUs. The noise-free QNN simulations utilize a custom noise-free simulator built upon the MindSpore framework (Xu et al., 2024), while classical neural networks (NNs) are implemented via the DeepXDE library (Lu et al., 2020).

For experimental validation on physical hardware, we utilized the IBM Quantum superconducting processor `ibm_fez` (accessing qubits q136 and q143). The quantum circuit transpilation, optimization, and error mitigation protocols (including Measurement Error Mitigation) were implemented using the Qiskit SDK (Javadi-Abhari et al., 2024).

5.1.2 Data Generation

Training and testing datasets are generated from mean-zero Gaussian random fields (GRF), $u \sim \mathcal{G}(0, k_l)$, characterized by the radial basis function (RBF) covariance kernel:

$$k_l(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2l^2}\right), \tag{34}$$

where $l > 0$ is the length-scale parameter controlling smoothness. The sampled functions u serve as inputs for the operator learning tasks. The target solutions are obtained using standard numerical solvers: the Runge–Kutta (4,5) method for ODE systems and a second-order finite difference scheme for PDEs. To construct the data pairs, for each input sample u, we randomly select P spatial locations in the output domain to evaluate the corresponding solution.

5.1.3 Training Configuration.

All reported results represent the mean and standard deviation (SD) computed over 5 independent trials with randomized data splitting and network initialization. To ensure a fair comparison, we constrain model complexity by setting the number of trainable parameters to approximately 1,200 for ODEs and 2,400 for PDEs. Optimization is performed using Adam (Kingma, 2014) with a learning rate of 10^{-4}. Detailed hyperparameters and architectural specifications are provided in Tables 1–3. All QNNs are implemented on a 5-qubit system. The Hamiltonian is defined as the sum of local Pauli-Z operators:

$$H = \sum_{i=1}^{5} \sigma_z^{(i)}. \tag{35}$$

We adopt this Hamiltonian as the default configuration due to its hardware-efficient implementation (native virtual Z-gates) and its prevalence as a standard benchmark in NISQ applications. Although theoretically a non-degenerate spectrum yields higher expressivity (as discussed in Sec. 5.4.1), this degenerate setup serves as a rigorous baseline to demonstrate the model's performance under restricted hardware conditions. Unless otherwise specified, these settings are kept consistent across all experimental trials.

Table 1: Default parameters for each problem, P denotes the number of sampling locations.

Case	u space	Sensors #	Train u #	Train P	Test u #	Test P	Batch Size	Iter. #
ODE	GRF($l = 0.2$)	100	1000	10	1000	100	100	100K
PDE	GRF($l = 0.2$)	100	1000	100	1000	1000	100	100K

Table 2: Hyperparameters for each network in the ODE case

Network Type	Branch		Trunk		Gates #	Param.#
	Depth h_b	Ansatz Depth l_b	Depth h_t	Ansatz Depth l_t		
QuanONet	20	2	20	2	1800	1201
TF-QuanONet	20	2	10	2	1350	1201
	Depth		Ansatz Depth			
HEA	40		2		1800	1201
TF-HEA	32		2		1440	1281
	Branch		Trunk			
	Depth	Width	Depth	Width		
DeepONet	2	10	2	10	—	1261
	Depth		Width			
FNN	2		10		—	1251

Table 3: Hyperparameters for each network in PDE case

Network Type	Branch		Trunk		Gates #	Param.#
	Depth h_b	Ansatz Depth l_b	Depth h_t	Ansatz Depth l_t		
QuanONet	40	2	40	2	3600	2401
TF-QuanONet	40	2	20	2	2700	2401
	Depth		Ansatz Depth			
HEA	80		2		2880	2401
TF-HEA	64		2		3600	2561
	Branch		Trunk			
	Depth	Width	Depth	Width		
DeepONet	3	15	3	15	—	2521
	Depth		Width			
FNN	3		16		—	2481

5.2 Comparison on Benchmark Problems

ODE Benchmarks.

- **Antiderivative Operator:** The operator $\mathscr{G} : u(x) \mapsto v(x)$ is defined by the relation:

$$\frac{\mathrm{d}v(x)}{\mathrm{d}x} = u(x), \quad v(0) = 0. \tag{36}$$

- **Homogeneous Operator:** The operator $\mathscr{G} : u(x) \mapsto v(x)$ governs the forced linear system:

$$\frac{\mathrm{d}v(x)}{\mathrm{d}x} = v(x) + u(x), \quad v(0) = 0. \tag{37}$$

- **Nonlinear Operator:** The operator $\mathscr{G} : u(x) \mapsto v(x)$ maps the forcing term to the solution of the cubic ODE:

$$\frac{\mathrm{d}v(x)}{\mathrm{d}x} = -v(x)^3 + u(x), \quad v(0) = 0. \tag{38}$$

PDE Benchmarks.

- **Diffusion-Reaction (D-R) System:** The operator $\mathscr{G} : u(x) \mapsto v(x,t)$ maps the source term u to the solution field:

$$\frac{\partial v}{\partial t} = D\frac{\partial^2 v}{\partial x^2} + kv^2 + u, \quad v(x,0) = 0, \tag{39}$$

  where the diffusion coefficient is $D = 0.01$ and the reaction rate is $k = 0.01$.

- **Advection Equation:** The operator $\mathscr{G} : u(x) \mapsto v(x,t)$ maps the initial condition $u(x)$ to the time-evolved state:

$$\frac{\partial v}{\partial t} = -c\frac{\partial v}{\partial x}, \quad v(x,0) = u(x), \tag{40}$$

  where the advection velocity is $c = 1$.

- **Darcy Flow:** The operator $\mathscr{G} : v|_{\partial D_2} \mapsto v$ maps the square boundary condition to the steady-state potential field:

$$-\nabla^2 v = f, \quad \text{with } f = -1, \tag{41}$$

  where the input is the boundary value $u = v|_{\partial D_2}$.

Experimental Setup. The computational domain for all independent variables is normalized to the unit interval $[0, 1]$. Specifically for the Darcy flow problem, the boundary input is discretized using sensors uniformly distributed along the square boundary with a resolution of 25, and the number of test sampling locations is set to $P = 500$.

5.2.1 Baselines

We comprehensively compare the proposed QuanONet and TF-QuanONet against state-of-the-art quantum and classical baselines:

- **Quantum Methods:** QNNs utilizing the Hardware-Efficient Ansatz (HEA) (Kandala et al., 2017) and the Trainable-Frequency HEA (TF-HEA) (Jaderberg et al., 2024).

- **Classical Methods:** Fully-connected Neural Networks (FNN) (Raissi et al., 2018) and the Deep Operator Network (DeepONet) (Lu et al., 2021). It is worth noting that we select the classical DeepONet as a performance proxy for the recently proposed QDeepONet (Xiao et al., 2024a). This is because Quantum DeepONet is designed primarily to accelerate the linear evaluation phase of DeepONet via quantum circuits, rather than to enhance its representational power. Consequently, the classical DeepONet represents the theoretical performance ceiling (ignoring quantum noise).

Table 4: Relative L_2 error comparison. The best is in gray and second best in lightgray.

Method	w_0	ODE		
		Antiderivative Operator	Homogeneous Operator	Nonlinear Operator
HEA	0.001	$71.789 \pm 0.804\%$	$81.474 \pm 0.943\%$	$69.913 \pm 1.239\%$
	0.01	$53.540 \pm 0.538\%$	$56.212 \pm 0.428\%$	$53.994 \pm 0.776\%$
	0.1	$40.233 \pm 0.505\%$	$45.838 \pm 0.695\%$	$38.440 \pm 0.801\%$
TF-HEA	0.001	$11.557 \pm 0.064\%$	$8.047 \pm 0.074\%$	$13.250 \pm 0.128\%$
	0.01	$11.634 \pm 0.091\%$	$8.040 \pm 0.061\%$	$13.292 \pm 0.154\%$
	0.1	$11.638 \pm 0.081\%$	$8.054 \pm 0.050\%$	$13.262 \pm 0.175\%$
QuanONet	0.001	$83.908 \pm 1.590\%$	$90.596 \pm 1.432\%$	$82.652 \pm 0.937\%$
	0.01	$46.370 \pm 1.394\%$	$51.806 \pm 1.633\%$	$46.618 \pm 1.283\%$
	0.1	$7.793 \pm 0.485\%$	$6.627 \pm 0.649\%$	$8.754 \pm 0.257\%$
TF-QuanONet	0.001	$2.376 \pm 0.166\%$	$2.335 \pm 0.202\%$	$2.911 \pm 0.267\%$
	0.01	$2.492 \pm 0.064\%$	$2.396 \pm 0.251\%$	$2.959 \pm 0.122\%$
	0.1	$2.160 \pm 0.173\%$	$2.196 \pm 0.157\%$	$2.635 \pm 0.230\%$
DeepONet		$5.162 \pm 1.044\%$	$3.538 \pm 0.635\%$	$6.099 \pm 1.014\%$
FNN		$6.233 \pm 1.437\%$	$4.699 \pm 1.615\%$	$7.211 \pm 2.096\%$
		PDE		
		D-R System	Advection Equation	Darcy Flow
HEA	0.001	$80.109 \pm 0.611\%$	$83.948 \pm 0.753\%$	$72.235 \pm 1.680\%$
	0.01	$79.150 \pm 0.630\%$	$76.873 \pm 0.516\%$	$61.325 \pm 0.515\%$
	0.1	$63.457 \pm 1.346\%$	$77.057 \pm 0.623\%$	$28.579 \pm 0.158\%$
TF-HEA	0.001	$12.874 \pm 0.577\%$	$66.017 \pm 10.712\%$	$6.962 \pm 0.127\%$
	0.01	$12.909 \pm 0.410\%$	$48.645 \pm 15.889\%$	$6.943 \pm 0.117\%$
	0.1	$12.547 \pm 0.241\%$	$32.432 \pm 0.606\%$	$6.651 \pm 0.178\%$
QuanONet	0.001	$80.473 \pm 0.777\%$	$91.023 \pm 1.124\%$	$81.755 \pm 0.942\%$
	0.01	$69.209 \pm 0.809\%$	$76.956 \pm 0.469\%$	$57.249 \pm 0.484\%$
	0.1	$31.099 \pm 0.770\%$	$50.424 \pm 1.825\%$	$11.977 \pm 0.051\%$
TF-QuanONet	0.001	$7.589 \pm 0.326\%$	$29.374 \pm 1.550\%$	$6.822 \pm 0.166\%$
	0.01	$7.733 \pm 0.433\%$	$27.028 \pm 2.995\%$	$6.914 \pm 0.207\%$
	0.1	$7.196 \pm 0.211\%$	$16.322 \pm 0.643\%$	$6.628 \pm 0.162\%$
DeepONet		$15.462 \pm 0.488\%$	$9.994 \pm 0.785\%$	$5.674 \pm 0.365\%$
FNN		$16.844 \pm 1.361\%$	$29.315 \pm 3.132\%$	$9.083 \pm 0.545\%$

5.2.2 Main Results (ODE & PDE Benchmarks)

As summarized in Table 4, in the ODE benchmarks, the standard QuanONet exhibits extreme sensitivity to the initialization scale w_0. While it achieves competitive performance at $w_0 = 0.1$, its accuracy collapses catastrophically as the frequency decreases to $w_0 = 0.001$, with errors surging to over 80%. This empirical evidence strongly supports our theoretical analysis regarding dimensional collapse: fixed low-frequency embeddings fail to span a sufficiently distinct spectral frame, degenerating the effective feature space. In sharp contrast, TF-QuanONet demonstrates remarkable robustness. Even under the adverse initialization of $w_0 = 0.001$, it maintains a low error rate ($\sim 2.4\%$), effectively matching its performance at optimal settings. This confirms that the TF strategy successfully breaks the

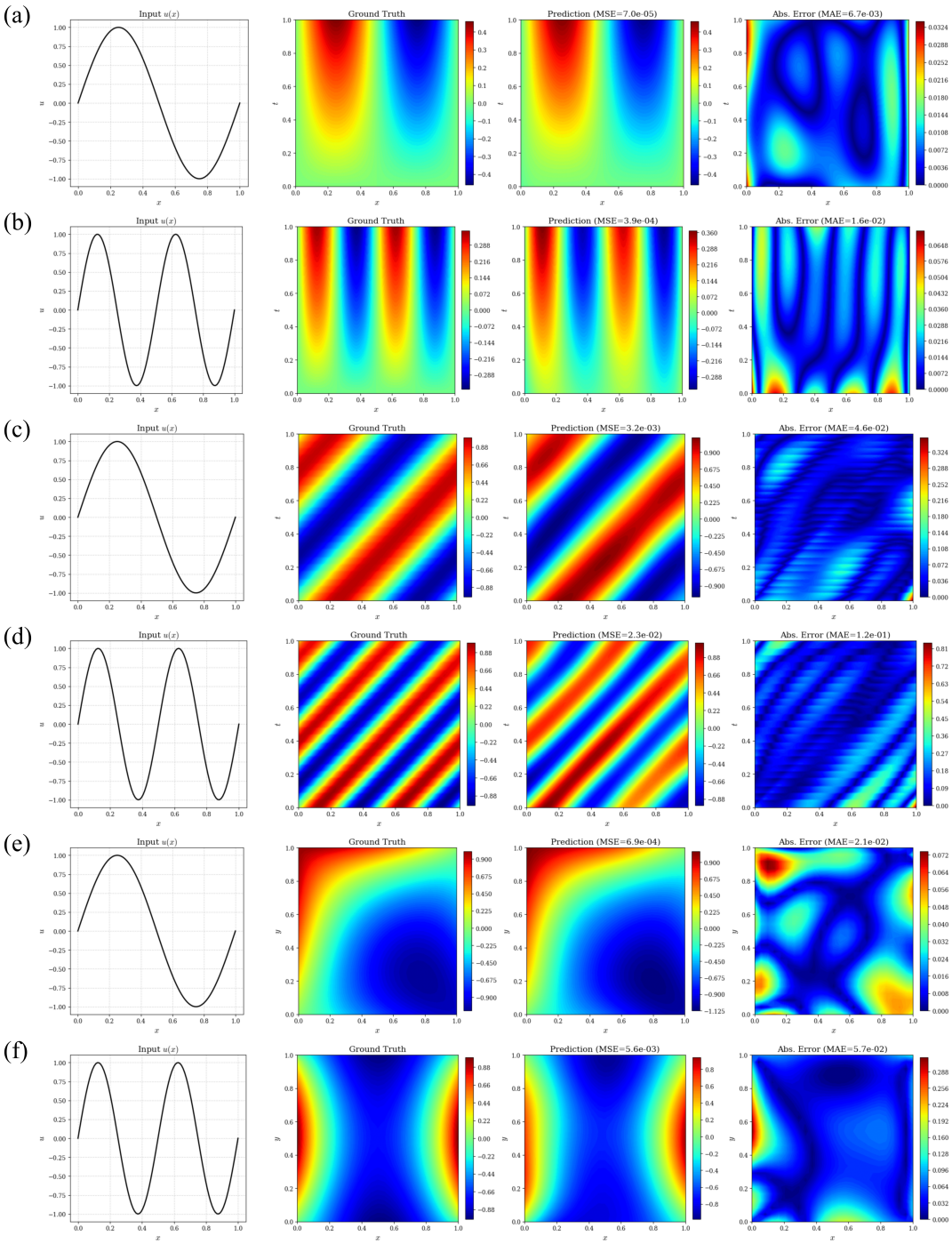

Figure 7: Visualization of TF-QuanONet predictions versus ground truth. The rows display results for the D-R System (a, b), Advection Equation (c, d), and Darcy Flow (e, f), corresponding to distinct input frequencies: $u(x) = \sin 2\pi x$ and $u(x) = \sin 4\pi x$.

spectral symmetry of the integer grid, autonomously recovering the $\mathcal{O}(p^2)$ frame capacity. Furthermore, TF-QuanONet consistently outperforms the classical DeepONet across all ODE tasks (e.g., 2.20% vs 3.54% on the Homogeneous operator), validating its superior parameter efficiency.

It is also worth noting the performance gap between TF-QuanONet and TF-HEA. Although both models utilize the trainable frequency mechanism, TF-QuanONet consistently surpasses TF-HEA. This indicates that the advantage stems not merely from the optimization trick, but from the QuanONet architecture itself—specifically, the quadratic kernel construction via the trace measurement of density matrices—which offers a fundamentally richer inductive bias than standard variational ansatzes.

For the PDE benchmarks, the results present a nuanced landscape. On the nonlinear D-R system, TF-QuanONet achieves an error of 7.20% compared to 15.46% for DeepONet and 16.84% for FNN. This suggests that for reaction-diffusion dynamics dominated by source terms, the quantum quadratic frame captures the operator mapping exceptionally well. However, for the Advection Equation and Darcy Flow, DeepONet retains the lead. Notably, in the Advection case, TF-QuanONet shows sensitivity to initialization again (29.37% at $w_0 = 0.001$ vs 16.32% at $w_0 = 0.1$). This implies that the transport dynamics in the Advection equation possess a broader and higher-frequency spectrum. While the TF strategy mitigates this, the model still benefits from higher initial frequencies to capture sharp wavefronts. We hypothesize that the classical DeepONet's advantage here stems from its massive parameter count, which allows it to brute-force the fitting of these high-frequency transport features, whereas the compact quantum model reaches its capacity limit.

5.2.3 Comparison with Fourier Neural Operators

To ensure a rigorous benchmarking against spectral-based architectures, we extend our evaluation to include the classical Fourier Neural Operator (FNO). A methodological alignment is requisite here; unlike QuanONet, which naturally accommodates unstructured sensor data, FNO-based architectures rely on FFT mechanisms that strictly mandate structured grids. Consequently, we standardize the input discretization to a uniform grid of $m = 100$ sensors for all models in this subsection to establish a unified evaluation framework compatible with FNO protocols.

A critical divergence from the previous ODE/PDE experiments lies in the model configuration. While we previously enforced strict parameter alignment ($\approx 1,200$), scaling FNO down to such a compact regime imposes severe architectural constraints, often compromising its spectral convolution efficacy. To address this, we adopt an asymmetric evaluation protocol designed to rigorously stress-test the quantum advantage. We configure the classical baselines (both FNO and DeepONet) with approximately $10,000$ parameters, whereas TF-QuanONet retains its highly compact configuration of roughly $1,200$ parameters. This setup grants the classical models nearly an order of magnitude advantage in parametric capacity. Despite this disparity, we maintain alignment on the effective latent dimension across architectures to ensuring that any performance differences reflect the efficiency of the representational frame rather than mere network width.

Table 5: Hyperparameters for each network.

Network Type	Branch		Trunk		Gates #	Param.#
	Depth h_b	Ansatz Depth l_b	Depth h_t	Ansatz Depth l_t		
TF-QuanONet	20	2	10	2	1350	1201
	Branch		Trunk			
	Depth	Width	Depth	Width		
DeepONet	4	32	4	32	—	9633
	Modes #	Width		Layers #		
FNO	15	14		3	—	10005

Table 6: Test error comparison on ODE benchmarks under asymmetric parameterization.

Method	Antiderivative Operator	Homogeneous Operator	Nonlinear Operator
TF-QuanONet	$3.127 \pm 0.292\%$	$2.722 \pm 0.117\%$	$5.191 \pm 0.454\%$
DeepONet	$6.330 \pm 0.197\%$	$6.982 \pm 1.421\%$	$4.040 \pm 0.446\%$
FNO	$5.924 \pm 1.436\%$	$3.684 \pm 0.443\%$	$3.952 \pm 0.351\%$

The experimental results, detailed in Table 6, reveal a distinct dichotomy that highlights both the efficiency and the capacity limits of the quantum architecture. On linear benchmarks (Antiderivative and Homogeneous operators), TF-QuanONet exhibits exceptional parameter efficiency. Despite operating with only $\sim 12\%$ of the parameter budget of the classical baselines ($1,201$ vs $\sim 10,000$), it achieves the lowest test errors (3.13% and 2.72%), significantly outperforming the over-parameterized FNO (5.92% and 3.68%). This confirms that for these operator classes, the implicit quadratic frame constructed by the density matrix provides a far more compact representation than the linear expansion used by classical spectral methods.

However, for the Nonlinear Operator, the ultra-compact TF-QuanONet (4.89%) is outperformed by the heavy classical baselines ($\approx 4.0\%$). This performance divergence is theoretically consistent with our frame analysis. Nonlinear operators typically induce target mappings with significantly higher spectral complexity and effective rank, necessitating a richer representational frame to be fully captured. While TF-QuanONet is structurally efficient, its absolute expressivity is ultimately bounded by the circuit depth and parameter count. In this highly nonlinear regime, the $1,200$-parameter configuration appears to reach a saturation point (underfitting), where the induced quantum frame struggles to span the complex target manifold. In contrast, the classical models leverage their massive parameter surplus to brute-force the approximation. This suggests that while QuanONet possesses superior information density, tackling highly nonlinear dynamics under such extreme compression requires scaling the quantum circuit depth beyond this minimal configuration.

5.3 Ablation Studies

5.3.1 Efficacy of Implicit Frame Construction

While the approximation error of DeepONet is theoretically lower-bounded by the representational capacity of its fixed latent dimension p (Lanthaler et al., 2022), QuanONet is hypothesized to access an implicit frame of size $\mathcal{O}(p^2)$ via the quadratic structure of the density matrix. To empirically validate this representational advantage, we compare the two architectures under a strictly constrained latent dimension of $p = 4$ (equivalent to 2 qubits).

To isolate the impact of this mechanism, we benchmark TF-QuanONet against DeepONet on the Antiderivative operator. We perform a sweep over various network depths and widths to ensure that any observed performance disparity is strictly attributable to the structural efficiency of the frame rather than mere parameter bottlenecks.

Table 7 and Fig. 8 reveal distinct saturation behaviors.

Crucially, to rule out under-parameterization as a confounding factor for the classical baseline, we extended the DeepONet ablation to an extreme regime, scaling the architecture to over 6.4×10^6 parameters. Despite this massive capacity—exceeding the QuanONet parameter count ($\sim 10^3$) by three orders of magnitude—the classical model remained trapped at a rigid error floor of approximately 5.5%. This saturation reflects the fundamental bottleneck of the linear frame, which cannot be overcome simply by increasing the width or depth of the neural networks.

In sharp contrast, TF-QuanONet consistently breaks through this error floor, achieving errors in the range of 2.5% with significantly fewer parameters. This empirical evidence substantiates our theoretical claim that the density matrix effectively constructs the quadratic terms $D_{ij}(\mathbf{y}) = t_i(\mathbf{y})t_j^*(\mathbf{y})$, thereby physically realizing a quadratic extension of the frame. This capability allows QuanONet to span a larger effective frame that remains inaccessible to classical models constrained to the linear capacity of the same latent dimension.

Notably, as shown in Fig. 8, DeepONet exhibits a substantial generalization gap—where the training error is significantly lower than the test error. This discrepancy indicates that the massive parameterization leads to overfitting. In sharp contrast, TF-QuanONet demonstrates a negligible generalization gap, with training and test errors tracking closely. This suggests that the quantum architecture possesses a strong intrinsic inductive bias that prevents overfitting, allowing it to generalize efficiently even with limited parameters.

To further probe the limits of operator approximation, we expanded the evaluation to a broader range of latent dimensions, sweeping p from 4 to 256. This scaling analysis, visualized in Fig. 9, reveals a distinct divergence in the optimization stability between the two architectures in high-dimensional regimes. Detailed experimental configurations are provided in Appendix D.

In the low-dimensional phase ($p = 4$), TF-QuanONet achieves lower reconstruction error, leveraging its high-capacity implicit feature map. As the dimension increases to $p = 8$, both architectures rapidly converge towards a shared intrinsic error floor of approximately 2×10^{-5}, a limit likely imposed by the encoding precision and residual approximation errors.

However, as the dimension scales further ($p > 32$), we observe a notable difference in consistency. While the classical DeepONet reaches the error floor, it exhibits visible stochastic

Table 7: Relative L_2 error comparison. The lighter gray represents smaller error (better performance).

(a) TF-QuanONet Performance

Trunk Depth h_t \ Branch Depth h_b	50	100	150	200
10	$4.524 \pm 0.960\%$	$3.280 \pm 0.061\%$	$3.602 \pm 0.176\%$	$3.551 \pm 0.302\%$
20	$3.805 \pm 0.672\%$	$3.137 \pm 0.226\%$	$3.187 \pm 0.111\%$	$3.052 \pm 0.123\%$
30	$3.018 \pm 0.143\%$	$2.842 \pm 0.350\%$	$2.959 \pm 0.193\%$	$2.981 \pm 0.448\%$
40	$3.105 \pm 0.151\%$	$2.944 \pm 0.358\%$	$2.639 \pm 0.394\%$	$2.607 \pm 0.377\%$
50	$3.072 \pm 0.111\%$	$2.673 \pm 0.178\%$	$2.642 \pm 0.491\%$	$2.617 \pm 0.368\%$
60	$3.091 \pm 0.225\%$	$2.509 \pm 0.407\%$	$2.490 \pm 0.220\%$	$2.516 \pm 0.293\%$
100	$2.800 \pm 0.341\%$	$2.235 \pm 0.181\%$	$2.478 \pm 0.450\%$	$2.512 \pm 0.273\%$
150	$3.332 \pm 0.617\%$	$2.517 \pm 0.404\%$	$2.381 \pm 0.488\%$	$2.728 \pm 0.558\%$
200	$3.309 \pm 0.566\%$	$2.833 \pm 0.645\%$	$2.705 \pm 0.272\%$	$2.447 \pm 0.213\%$
300	$3.065 \pm 0.375\%$	$2.474 \pm 0.097\%$	$2.968 \pm 0.433\%$	$2.601 \pm 0.245\%$

(b) DeepONet Performance

B&T Width \ B&T Depth	2	3	4	5
4	$9.840 \pm 1.944\%$	$7.211 \pm 1.894\%$	$6.982 \pm 0.389\%$	$8.946 \pm 1.803\%$
8	$6.222 \pm 0.278\%$	$5.769 \pm 0.139\%$	$5.443 \pm 0.447\%$	$5.137 \pm 0.785\%$
16	$5.704 \pm 0.150\%$	$5.557 \pm 0.096\%$	$5.648 \pm 0.081\%$	$5.141 \pm 0.921\%$
32	$5.740 \pm 0.135\%$	$5.608 \pm 0.073\%$	$5.580 \pm 0.120\%$	$5.056 \pm 0.992\%$
64	$5.888 \pm 0.104\%$	$5.778 \pm 0.124\%$	$5.892 \pm 0.142\%$	$6.096 \pm 0.113\%$
128	$6.028 \pm 0.189\%$	$5.905 \pm 0.081\%$	$6.075 \pm 0.106\%$	$6.521 \pm 0.200\%$
256	$6.066 \pm 0.163\%$	$5.969 \pm 0.226\%$	$5.886 \pm 0.134\%$	$6.242 \pm 0.247\%$
512	$6.012 \pm 0.077\%$	$5.512 \pm 0.090\%$	$5.629 \pm 0.096\%$	$5.915 \pm 0.234\%$
1024	$6.041 \pm 0.238\%$	$5.499 \pm 0.099\%$	$5.599 \pm 0.082\%$	$5.703 \pm 0.048\%$

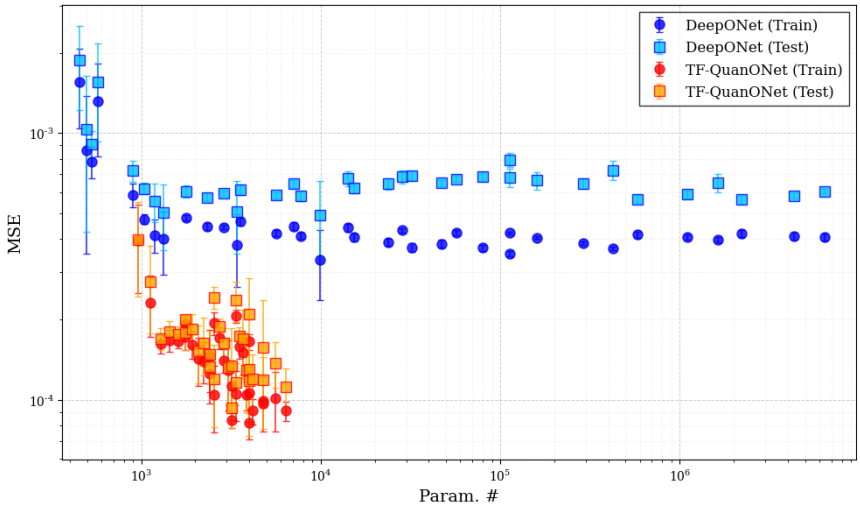

Figure 8: Test error of TF-QuanONet and DeepONet versus parameter counts. While TF-QuanONet (orange) continues to improve with model complexity, DeepONet (blue) exhibits a clear saturation behavior.

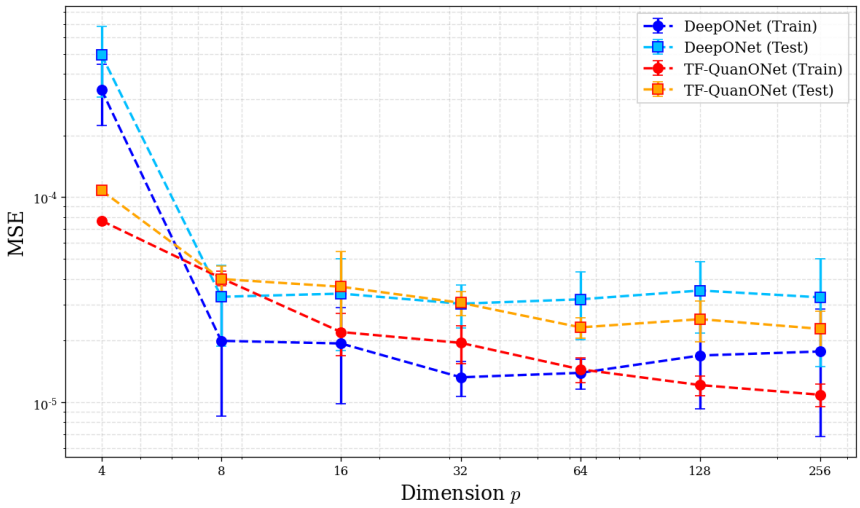

Figure 9: Scaling behavior of Train/Test MSE with respect to the latent dimension p, sweeping from $p = 4$ to $p = 256$ (corresponding to 2–8 qubits). The curves report the optimal performance achieved across extensive network size configurations.

fluctuations across different random seeds, with test errors varying between 2×10^{-5} and 4×10^{-5}. This observation indicates that in high-dimensional spaces, the classical baseline becomes increasingly sensitive to initialization, potentially struggling to consistently locate the global optimum within the optimization landscape.

In contrast, TF-QuanONet displays remarkable stability. Even at $p = 256$, the quantum model consistently converges to the error lower bound with negligible variance across runs. While the precise mechanism warrants further theoretical investigation, this behavior suggests that the TF-QuanONet architecture may possess a more robust optimization landscape or inherent structural regularization that mitigates the optimization variance typically observed in high-dimensional operator learning.

5.3.2 Impact of Circuit Architecture: Width and Depth

Although theoretical analysis suggests that increasing system size enhances expressivity, practical quantum model training is often constrained by optimization challenges. We investigate the trade-off between circuit complexity—defined by width (qubit count) and depth (h_b, h_t)—and approximation accuracy. Table 8 summarizes the relative errors on the Antiderivative operator across varying configurations.

We observe that increasing the depth of both the Branch (h_b) and Trunk (h_t) layers consistently improves model performance. For a fixed qubit count, deeper circuits effectively reduce the approximation error; for instance, with 5 qubits, increasing depths from $(h_b = 20, h_t = 10)$ to $(40, 40)$ reduces the error from 2.41% to 1.64%. This trend confirms that sufficient circuit depth is critical for capturing the complex frequency spectrum of the target operator, aligning with our theoretical error bounds.

However, the impact of circuit width (number of qubits) is governed by two competing mechanisms. Increasing the qubit count expands the dimension of the latent Hilbert space,

Table 8: Ablation study on circuit architecture w.r.t. the Relative L_2 error (Parameter #). h_b and h_t are depths of Branch and Trunk layers, respectively.

Qubit #	h_b \ h_t	10	20	30	40
		Rel. L_2 Error (Param.#)			
2	50	$4.524 \pm 0.960\%$ (961)	$3.805 \pm 0.672\%$ (1121)	$3.018 \pm 0.143\%$ (1281)	$3.105 \pm 0.151\%$ (1441)
	100	$3.280 \pm 0.061\%$ (1761)	$3.137 \pm 0.226\%$ (1921)	$2.842 \pm 0.350\%$ (2081)	$2.944 \pm 0.358\%$ (2241)
5	20	$2.414 \pm 0.164\%$ (1201)	$2.252 \pm 0.106\%$ (1601)	$2.212 \pm 0.078\%$ (2001)	$1.879 \pm 0.128\%$ (2401)
	40	$2.200 \pm 0.078\%$ (2001)	$1.761 \pm 0.147\%$ (2401)	$1.613 \pm 0.121\%$ (2801)	$1.638 \pm 0.211\%$ (3201)
10	10	$5.754 \pm 0.236\%$ (1601)	$3.744 \pm 0.268\%$ (2401)	$3.076 \pm 0.086\%$ (3201)	$2.597 \pm 0.216\%$ (4001)
	20	$3.413 \pm 0.234\%$ (2401)	$2.672 \pm 0.194\%$ (3201)	$2.306 \pm 0.201\%$ (4001)	$2.045 \pm 0.101\%$ (4801)

theoretically reducing reconstruction error via richer feature representation. Conversely, larger systems significantly increase the complexity of the quantum optimization landscape. This can exacerbate training difficulties, such as the barren plateau problem, potentially negating the gains in representational capacity. Our results reflect this trade-off: the 5-qubit model achieves the optimal balance with errors around 1.6%, whereas the 10-qubit model exhibits performance degradation (errors rising to 2.05%–5.75%) under comparable parameter scales. Consequently, selecting the qubit count requires a careful balance between maximizing latent space dimensionality and ensuring trainability.

5.4 Impact of Hamiltonian Design

5.4.1 Spectral Degeneracy and Manifold Dimension

In the QuanONet framework, the Hamiltonian serves as the measurement operator, and its spectral structure fundamentally dictates the model's expressivity. The Branch layers encode input information by mapping it into a unitary orbit $\mathcal{M}_{\mathbf{\Lambda}}$ generated by the Hamiltonian's spectrum. The dimension of this orbit is given by $\dim(\mathcal{M}_{\mathbf{\Lambda}}) = p^2 - \sum_{j=1}^{q} n_j^2$, where n_j is the multiplicity of each eigenvalue. A non-degenerate spectrum (where all eigenvalues are distinct) maximizes this dimension to $p^2 - p$, whereas degeneracy collapses the orbit and imposes rigid constraints on the model's capacity.

To empirically validate this hypothesis, we conducted experiments on the Antiderivative operator using a 2-qubit system ($p = 4$) with a fixed spectral radius, and sizes are all set to $h_b = 50, h_t = 50$. Four Hamiltonians with varying degrees of degeneracy were tested:

$$\begin{aligned} \mathbf{\Lambda}_1 &= \mathrm{diag}[-5, 5, 5, 5], \quad \mathbf{\Lambda}_2 = \mathrm{diag}[-5, -5, -5, 5], \\ \mathbf{\Lambda}_3 &= \mathrm{diag}[-5, 0, 0, 5], \quad \mathbf{\Lambda}_4 = \mathrm{diag}[-5, -2.5, 2.5, 5], \end{aligned} \tag{42}$$

ranging from a highly degenerate spectrum (dimension 6) to a fully non-degenerate spectrum (dimension 12).

As illustrated in Fig. 10, model performance generally correlates with manifold dimension, with the non-degenerate $\mathbf{\Lambda}_4$ achieving optimal accuracy. However, $\mathbf{\Lambda}_2$ ($\mathrm{diag}[-5, -5, 5, 5]$) presents an exception, underperforming the lower-dimensional $\mathbf{\Lambda}_1$. This anomaly underscores that the Hamiltonian spectrum governs not only the manifold dimension (via degeneracy) but also its intrinsic geometric structure, as hinted by the visualization in Fig. 5.

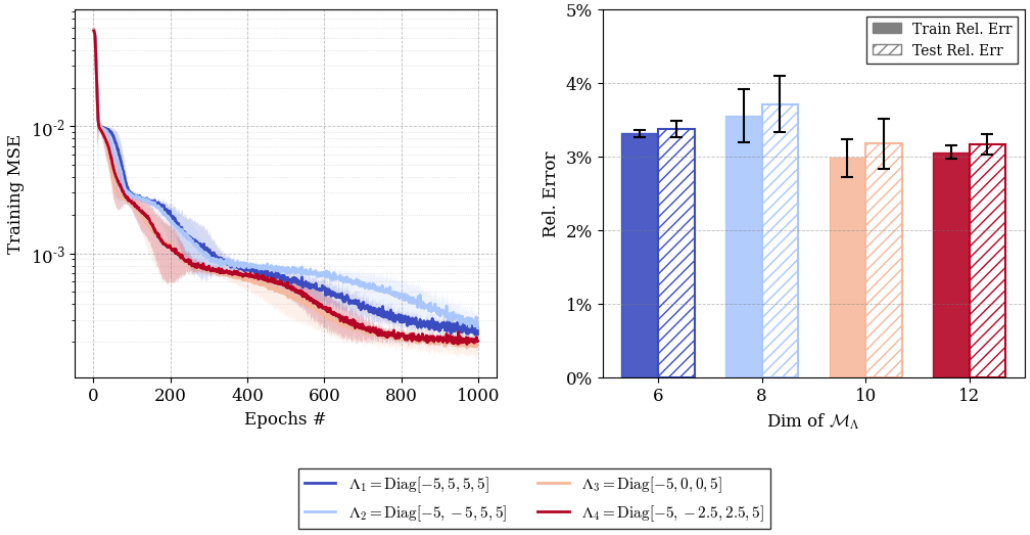

Figure 10: Test error of TF-QuanONet on the Antiderivative operator using Hamiltonians with different spectra. The legend indicates the eigenvalues. The effective manifold dimensions are $\dim(\mathcal{M}_{\mathbf{\Lambda}}) = 6, 8, 10, 12$, respectively. Performance improves as the manifold dimension increases.

The specific configuration of $\mathbf{\Lambda}_2$ likely induces structural constraints that hinder effective approximation, suggesting a complex interplay between spectral properties and model expressivity beyond simple dimensionality maximization.

5.4.2 Pauli Terms and Spectral Radius

We further investigate the impact of the Hamiltonian's spectral properties beyond degeneracy. As shown in Fig. 11, the model exhibits invariance to the specific choice of Pauli basis (e.g., $\sigma_x, \sigma_y, \sigma_z$), provided the eigenspectrum is preserved. In contrast, the spectral radius $\rho(\mathbf{\Lambda}) = \max_i |\lambda_i|$ emerges as a decisive hyperparameter. Theoretically, $\rho(\mathbf{\Lambda})$ bounds the feasible region of the achievable coefficients.

Empirically, we observe a distinct threshold effect: when $\rho(\mathbf{\Lambda})$ is insufficient to enclose the target function's codomain, accuracy saturates due to geometric obstruction (underfitting). This observation aligns perfectly with the geometric landscape depicted in Fig. 5: increasing the spectral radius effectively inflates the observable manifold, ensuring it volumetrically encompasses the target coefficient space required by the operator mapping. Conversely, once $\rho(\mathbf{\Lambda})$ covers the target range, further expansion yields negligible gains as the model adapts via unitary rotations. This confirms that macroscopic spectral properties—specifically, a sufficient magnitude—supersede microscopic basis choices in ensuring universality.

5.5 Real-Device Deployment

The deployment of quantum algorithms on NISQ devices necessitates a rigorous evaluation of their resilience against inherent hardware noise. We systematically investigate the ro-

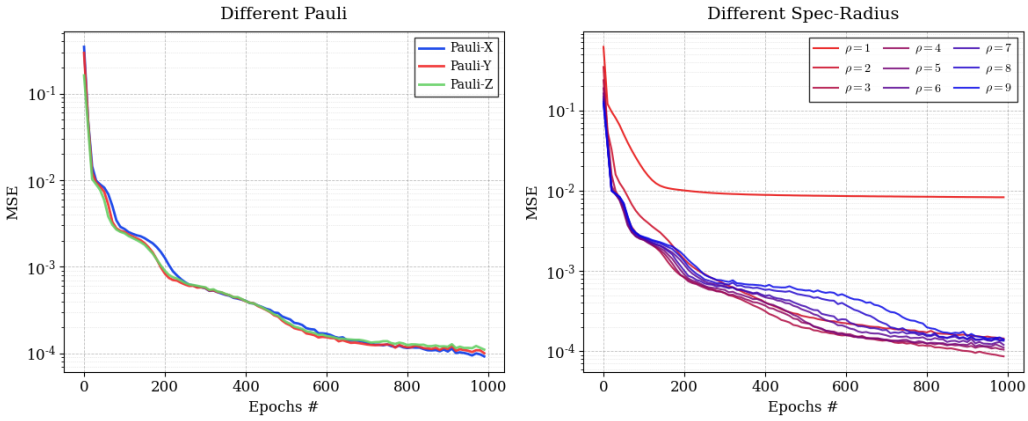

Figure 11: Error curve of TF-QuanONet versus Hamiltonian spectral radius for the Antiderivative operator. Here, $\rho = 7$ denotes a scaled Hamiltonian $H = \frac{7}{5}\sum_{i=1}^{5}\sigma_z^{(i)}$. The error plateaus once the radius sufficiently covers the target function's range.

bustness of TF-QuanONet using the real superconducting quantum processor described in Sec. 5.1.1. To accommodate the limited coherence time of current qubits, we designed an ultra-shallow circuit configuration with depth parameters $h_b = h_t = 5$ and ansatz repetition $l_b = l_t = 1$ for the Antiderivative operator ($m = 10$).

We evaluated the model on linear ($u(x) = x$) and trigonometric ($u(x) = \cos(\pi x)$) inputs. The inference process utilized 10,000 shots per evaluation. As detailed in Table 9, the Qiskit transpiler mapped the logical circuit to the native gate set $(R_z, \sqrt{X}, CZ)$, resulting in a physical depth of 109. This represents a manageable overhead compared to the logical depth of 60, ensuring the circuit execution remains within the coherence window.

Quantitatively, the transition from noise-free simulation to real hardware incurs a performance penalty of approximately one to two orders of magnitude in MSE (Table 9), shifting relative errors from the $\sim 5\%$ regime to $\sim 30\%$–40%. Despite this numerical gap, the qualitative performance remains robust. As illustrated in Fig. 12, the hardware inference successfully reconstructs the characteristic curvature and trends of the target solutions, rather than degenerating into stochastic noise. These results empirically validate that TF-QuanONet, equipped with a hardware-efficient topology, maintains functional expressivity on NISQ devices without requiring heavy error correction overhead.

Nevertheless, we must objectively acknowledge a critical limitation. For the specific domain of differential equation solving—where high numerical precision is often non-negotiable—the noise-induced error floor remains a formidable barrier to practical deployment. While the TF-QuanONet successfully learns the solution topology, the quantitative deviation poses a fatal bottleneck for engineering-grade applications. Consequently, bridging the gap from qualitative capture to precision solving will inevitably require the maturation of quantum error correction (QEC) and fault-tolerant protocols in future hardware generations.

Table 9: Performance comparison between noise-free simulator and real-device (`ibm_fez`), alongside detailed circuit compilation statistics.

$u(x)$	Metric	Simulator	Real-device
x	MSE	1.534×10^{-4}	8.222×10^{-3}
	Rel. L_2 Error	5.498%	40.247%
$\cos \pi x$	MSE	7.400×10^{-5}	5.190×10^{-3}
	Rel. L_2 Error	3.841%	32.168%

Gate Complexity	Metric	Logical	Physical
Compilation	Circuit Depth	60	109
	Gate Breakdown	$R_y(40), R_x(20)$ $R_z(20), CX(20)$	$R_z(92), \sqrt{X}(82)$ $CZ(20)$
	Total Count	100	194

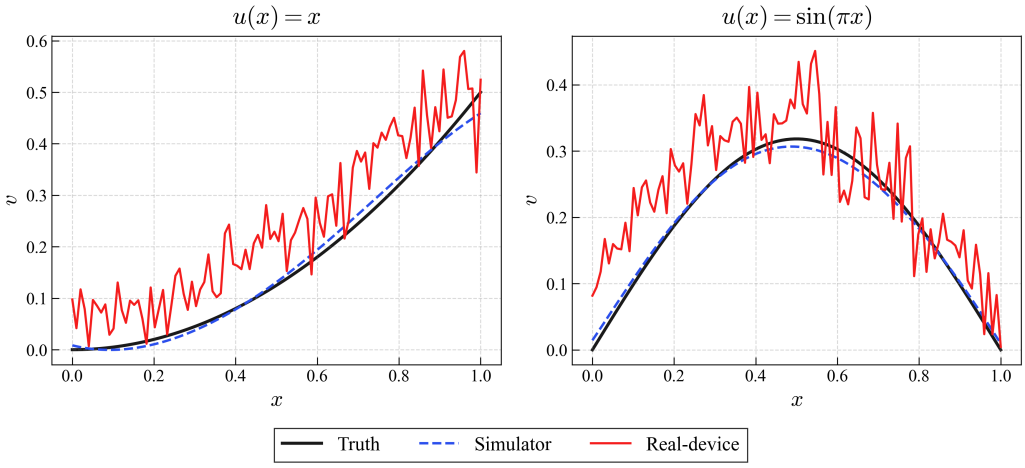

Figure 12: Inference trajectories on real hardware. Despite the quantitative noise floor, the TF-QuanONet successfully captures the functional curvature of the target solutions.

6 Conclusion and Outlook

In this work, we have established a comprehensive framework for quantum operator learning, bridging the gap between rigorous approximation theory and practical circuit implementation. The key contributions and findings are summarized as follows:

Theoretical Foundations and Implicit Expressivity. We derived a constructive proof for universal approximation, rigorously establishing an expressivity lower bound of $\mathcal{O}(p)$. Crucially, beyond this guarantee, we identified and empirically validated that the trace measurement protocol implicitly constructs a quadratic frame of size $\mathcal{O}(p^2)$. This structural expansion allows QuanONet to surpass the representational limits of classical linear baselines, yielding superior parameter efficiency and generalization stability.

Algorithmic Robustness and Efficiency. We addressed the spectral redundancy inherent in Fixed-Frequency embeddings by introducing the Trainable-Frequency (TF-

QuanONet) paradigm. By breaking spectral symmetries, this strategy ensures that the implicit frame frequencies are non-commensurate, allowing the model to autonomously optimize its spectral covering. Consequently, it yields consistently superior accuracy across diverse differential equation benchmarks compared to standard variational ansatzes.

Real-Device Viability and Structural Stability. Both simulation and physical deployment on IBM Quantum processors validate the feasibility of TF-QuanONet on near-term hardware. Despite the inevitable quantitative error floor imposed by NISQ noise, the model successfully recovers the characteristic functional topology of the solution operators. This capability to capture fundamental physical laws under noisy conditions validates the architecture's hardware compatibility, marking a crucial step toward practical quantum scientific computing.

Transcending specific benchmarks, this work advocates for a conceptual shift in QNN design: moving from the standard paradigm of state vector preparation to the direct parameterization of unitary operators on the manifold. While this perspective highlights the need for further theoretical development regarding parameterized unitaries, the QuanONet architecture establishes a versatile backbone to navigate these frontiers. By facilitating the design of functional quantum layers tailored to specific operator classes, our framework lays the groundwork for scaling quantum neural operators toward complex fluid dynamics and multi-physics simulations, representing a vital step toward unlocking practical quantum advantage in scientific machine learning.

Acknowledgments

This work was partly supported by National Natural Science Foundation of China (92370201). Acknowledgment goes to Tengyue Zhang, Yingjie Xiong and Can Chen, for their assistance in completing the experiments.

Appendix A. Nomenclature

Table 10: Nomenclature and definitions of symbols used in this paper.

Symbol	**Description**
Sets and Spaces	
$\mathbb{R}^d, \mathbb{C}^d$	Real and Complex vector spaces of dimension d
D_1, D_2	Compact spatial domains for input and output functions
$\mathrm{C}(D)$	Banach space of continuous functions on domain D
K_1, K_2	Compact sets of input and output functions ($K_1 \subset \mathrm{C}(D_1)$, $K_2 \subset \mathrm{C}(D_2)$)
$\mathcal{H}$	Finite-dimensional quantum Hilbert space (isomorphic to $\mathbb{C}^p$)
$\mathcal{L}(\mathcal{H})$	Space of linear operators acting on $\mathcal{H}$
$\mathcal{M}_{\mathbf{\Lambda}}$	Unitary orbit generated by the fixed spectrum $\mathbf{\Lambda}$
Functions and Variables	
$u(\mathbf{x})$	Input function (e.g., initial condition or source term)
$v(\mathbf{y})$	Output function (solution to the differential equation)
$\mathbf{x}, \mathbf{y}$	Spatial coordinates in the input domain D_1 and output domain D_2
$\tilde{\mathbf{u}}$	Discretized input vector $[u(\mathbf{x}_1), \ldots, u(\mathbf{x}_m)]^\top$
$\mathscr{G}$	Target nonlinear continuous operator $\mathscr{G} : K_1 \to K_2$
Network Architecture	
$\mathscr{D}, \mathscr{Q}$	Operator mappings for DeepONet and QuanONet, respectively
$\mathscr{E}$	Encoder mapping input functions to finite-dimensional vectors
$\mathscr{A}, \mathscr{A}_Q$	Approximator mappings for DeepONet and QuanONet respectively
$\mathscr{R}, \mathscr{R}_Q$	Reconstructor mappings for DeepONet and QuanONet respectively
$\mathbf{b}(u), \mathbf{t}(\mathbf{y})$	Amplitude vectors of the Branch and Trunk quantum states
p	Latent dimension (Hilbert space dimension $\dim(\mathcal{H}) = 2^n$, where n is qubit count)
m	Number of sensor locations for input discretization
h_b, h_t	Depth (repetitions) of Branch and Trunk quantum circuits
l_b, l_t	Depth of the variational ansatz in Branch and Trunk layers
$\boldsymbol{\theta}$	Learnable parameters of the quantum circuits ($\boldsymbol{\theta}^b, \boldsymbol{\theta}^t$)
b_0	Trainable scalar bias added to the inner product output
$\boldsymbol{\omega}, \boldsymbol{\kappa}$	Trainable frequency weights and biases
Quantum Operators and States	
$\mathbf{M}(u)$	Branch-Modulated Observable encoding input u
$\mathbf{D}(\mathbf{y})$	Trunk-Density Matrix encoding coordinate $\mathbf{y}$
H	Base measurement Hamiltonian (e.g., sum of Pauli-Z operators)

Symbol	Description
$\mathbf{\Lambda}$	Diagonal matrix of eigenvalues of H, $\mathbf{\Lambda} = \mathrm{diag}[\lambda_1, \ldots, \lambda_p]$
$\rho(\mathbf{\Lambda})$	Spectral radius of the matrix $\mathbf{\Lambda}$
$\lvert\phi_i\rangle, \lambda_i$	The i-th eigenvector and eigenvalue of the Hamiltonian
$U_{\mathrm{BNC}}, U_{\mathrm{TNK}}$	Unitary evolution operators for Branch and Trunk circuits
$\mathbf{B}(u)$	Matrix representation of the Branch unitary $U_{\mathrm{BNC}}^{\dagger}$
$\langle\cdot,\cdot\rangle_F$	Frobenius inner product, defined as $\mathrm{Tr}[\mathbf{A}^{\dagger}\mathbf{B}]$

Appendix B. Further Related Works

The landscape of quantum algorithms for solving Partial Differential Equations (PDEs) has evolved rapidly, generally categorized into three distinct paradigms: fault-tolerant quantum simulation, variational quantum algorithms (VQAs), and the emerging field of quantum neural operators.

B.1 Quantum Simulation and Linear Algebra Subroutines

The theoretical foundation for quantum PDE solving was established by the Harrow-Hassidim-Lloyd (HHL) algorithm (Harrow et al., 2009), which proposed a method to solve linear systems of equations $\mathbf{Ax} = \mathbf{b}$ with complexity scaling logarithmically with the system size $\mathcal{O}(\log N)$, offering an exponential speedup over classical conjugate gradient methods. However, HHL requires deep quantum circuits and efficient Quantum Random Access Memory (QRAM), limiting its immediate applicability in the NISQ era. To address general linear PDEs—including non-Hermitian and dissipative systems like the heat or convection-diffusion equations— Jin and Liu (2022) introduced the Schrödingerization framework. This technique employs a warped phase transformation to embed non-unitary dynamics into higher-dimensional unitary evolution, allowing standard Hamiltonian simulation methods to be applied to a wider class of PDEs. More recently, this approach has been extended to nonlinear equations via the HAM-Schrödingerization method (Liao, 2024), which combines homotopy analysis with quantum simulation to iteratively solve linearized sub-problems.

B.2 Variational Quantum Algorithms (VQAs)

Acknowledging the hardware constraints of the NISQ era, research has shifted towards hybrid quantum-classical approaches. The Variational Quantum Linear Solver (VQLS) (Bravo-Prieto et al., 2019) replaces the deep phase estimation circuits of HHL with a shallow, parameterized ansatz, optimizing a local cost function to approximate the solution of linear systems. In the context of differential equations, Differentiable Quantum Circuits (DQC) (Kyriienko et al., 2021) have emerged as a quantum analogue of PINNs. These methods utilize the parameter-shift rule to analytically compute gradients of quantum circuits, enabling the incorporation of PDE residuals directly into the loss function. While VQAs significantly reduce circuit depth, they face challenges related to trainability, such as the barren plateau problem, where gradients vanish exponentially with system size.

B.3 Detailed Comparison with QDeepONet

To explicitly contextualize the proposed architecture, we highlight the critical distinctions between QuanONet and the recently proposed QDeepONet (Xiao et al., 2024a). First, regarding structural design, QuanONet is a native, end-to-end quantum framework in which both feature mapping and nonlinear transformations are implemented on quantum circuits. This allows for potential quantum-aware training. In contrast, QDeepONet adopts a hybrid paradigm that restricts quantum processing solely to accelerating linear matrix multiplications during inference, while relying on classical processors for nonlinear activation functions and the entire training process. Second, in terms of resource efficiency, QuanONet leverages amplitude embedding principles to compress the latent space, requiring only $\mathcal{O}(\log p)$ qubits to represent a network width of p. Conversely, QDeepONet employs unary encoding, which necessitates a qubit count scaling as $\mathcal{O}(p)$, thereby imposing severe hardware constraints for wide networks. Finally, the nature of the quantum advantage differs fundamentally. QDeepONet aims for computational speedup during the evaluation pass but offers no theoretical gain in approximation accuracy relative to its classical counterpart (and may suffer from noise-induced degradation). QuanONet, however, exploits the density matrix to implicitly construct a quadratic frame of size $\mathcal{O}(p^2)$ from $\mathcal{O}(p)$ latent dimension. This provides a rigorous representational advantage, enabling superior approximation accuracy for complex operators, as summarized in Table 11.

Table 11: Structural and functional comparison between QDeepONet and our proposed QuanONet. Here p denotes the width of the classical network counterpart.

Feature	QDeepONet (Xiao et al., 2024a)	QuanONet (Ours)
Architecture Type	Hybrid (Quantum Linear + Classical Non-linear)	Pure Quantum (End-to-End Circuit)
Qubit Complexity	$\mathcal{O}(p)$ (Orthogonal Layers)	$\mathcal{O}(\log p)$ (Amplitude Embedding)
Training & Inference	Classical Training; Hybrid Inference	Quantum Training & Inference
Frame Construction	Explicit Linear Frame ($\mathcal{O}(p)$)	Implicit Quadratic Frame ($\mathcal{O}(p^2)$)
Primary Advantage	Evaluation Speedup	Expressivity & Accuracy

B.4 Theoretical Connection to FNO

Recent theoretical advancements by Kovachki et al. (2021) have established a unifying perspective on operator learning, demonstrating that the FNO can be rigorously interpreted as a specialized instance of the DeepONet architecture. In this framework, the FNO corresponds to a DeepONet where the Trunk Net is structurally fixed to approximate a trigonometric basis, effectively performing a kernel convolution in the frequency domain. This insight invites a theoretical comparison regarding how frame functions are constructed in each architecture. The standard FNO paradigm explicitly parameterizes the integral operator using the FFT. By operating on a discrete integer lattice $\mathbb{Z}^d$ and applying a frequency truncation determined by a fixed bandwidth parameter, FNO efficiently learns the coefficients of standard harmonic modes e^{iky}. This design acts as a rigid spectral filter, prioritizing low-frequency modes to achieve rapid convergence for operators dominated by smooth dynamics but limiting the basis to a fixed harmonic grid. In contrast,

TF-QuanONet approaches spectral representation through a continuous mechanism governed by quantum evolution. Rather than pre-defining a fixed spectral grid, the Trunk layers generate encoding functions via the evolution of a state vector under a parameterized Hamiltonian. A distinct feature of this quantum architectural design is the incorporation of the Trainable-Frequency strategy. By treating embedding frequencies as continuous learnable parameters rather than fixed integers, the model retains the flexibility to adjust its spectral alignment during training. This mechanism allows the architecture to represent frequencies that may be non-commensurate or lie off the standard integer lattice. Consequently, while both architectures leverage spectral principles to capture global correlations, they diverge in their basis construction strategy: FNO employs a fixed, discrete trigonometric basis optimized for computational speed via FFT, whereas TF-QuanONet synthesizes an adaptive spectral representation driven by the continuous parameters of the quantum circuit.

Table 12: Comparison of spectral handling between FNO and TF-QuanONet.

Feature	FNO (Li et al., 2021)	QuanONet (Ours)
Trunk Architecture	Fixed Trigonometric Basis	Quantum State Evolution
Spectral Domain	Discrete Integer Lattice ($\mathbb{Z}^d$)	Continuous Real Spectrum ($\mathbb{R}^d$)
Frequency Handling	Fixed Modes (Truncated)	Learnable Parameters (Adaptive)
Primary Mechanism	High-frequency Filtering / FFT	Spectral Alignment / Hamilton Evolution

B.5 Connections with Classical Machine Learning Representations

The implicit quadratic frame construction, utilized here to enhance the expressivity of quantum neural operators, exhibits fundamental structural parallels with established classical machine learning paradigms designed for high-order feature interaction. Mathematically, the utilization of the Frobenius inner product between a modulated observable and a rank-1 density matrix formally implements a homogeneous polynomial kernel of degree two, implicitly mapping latent representations into a tensor product space of size $\mathcal{O}(p^2)$ without explicit coordinate expansion (Schölkopf and Smola, 2002). This reliance on the outer product structure to aggregate coordinate information establishes a direct methodological correspondence to bilinear pooling techniques in computer vision (Lin et al., 2015), where the density matrix functions as a global covariance descriptor capturing second-order statistical dependencies among latent state amplitudes. Furthermore, the operational form of the interaction mirrors linear attention mechanisms (Katharopoulos et al., 2020), as the density matrix serves as a compressed context representation that decouples input and output embeddings while preserving dense connectivity via the spectral trace operation. Finally, the introduction of the Trainable-Frequency strategy to resolve spectral dimensional collapse aligns with the latent vector optimization in Factorization Machines (Rendle, 2010); specifically, learning continuous embedding parameters ensures the linear independence of pairwise spectral differences, thereby preventing degeneracy and maximizing the expressivity of the quadratic frame.

Appendix C. Proofs of Quantum Universal Approximation Theorem

In this appendix, we provide rigorous proofs for the approximation capabilities of the proposed quantum neural operator architectures.

C.1 Mathematical Preliminaries

We adopt the notation from the main text. Let $D_1 \subset \mathbb{R}^{d_1}$ and $D_2 \subset \mathbb{R}^{d_2}$ be compact spatial domains. Let $K_1 \subset \mathrm{C}(D_1)$ be a compact set of input functions. The target operator is a continuous mapping $\mathscr{G} : K_1 \to \mathrm{C}(D_2)$.

We invoke the classical universal approximation theorem for nonlinear operators, established by Chen and Chen (1995), which serves as the foundation for our quantum extensions.

Theorem 1 (Universal Approximation of DeepONet) *Let K_1 be a compact subset of a Banach space of input functions, $D_2 \subset \mathbb{R}^{d_2}$ be a compact domain, and $\mathscr{G} : K_1 \to \mathrm{C}(D_2)$ be a continuous nonlinear operator. For any $\epsilon > 0$, there exists a Deep Operator Network $\mathscr{D}$ parameterized by a latent dimension p such that:*

$$\sup_{u \in K_1, \mathbf{y} \in D_2} \left| \mathscr{G}(u)(\mathbf{y}) - \sum_{k=1}^{p} b_k(u) t_k(\mathbf{y}) \right| < \epsilon. \tag{43}$$

This theorem implies that any continuous operator can be approximated by a finite dot product of a vector-valued functional $\mathbf{b}(u)$ and a vector-valued function $\mathbf{t}(\mathbf{y})$ in an Euclidean space $\mathbb{R}^p$.

C.2 Proof of Theorem 2

Here we demonstrate that the Euclidean dot product structure can be faithfully embedded into the inner product of normalized quantum states.

Proof First, we define the scaling factors based on the supremum of the Euclidean norms over the compact domains:

$$B := \sup_{u \in K_1} \|\mathbf{b}(u)\|_2, \quad T := \sup_{\mathbf{y} \in D_2} \|\mathbf{t}(\mathbf{y})\|_2. \tag{44}$$

Since K_1 and D_2 are compact and the mappings are continuous, these suprema are finite. By the assumption of non-trivial norms, we have $B > 0$ and $T > 0$. Let the global scaling constant be $\lambda = B \cdot T$.

We define the normalized, dimensionless coefficient functions for $k = 1, \ldots, p$:

$$\tilde{b}_k(u) = \frac{b_k(u)}{B}, \quad \tilde{t}_k(\mathbf{y}) = \frac{t_k(\mathbf{y})}{T}. \tag{45}$$

From the definition of B, it follows that $\sum_{k=1}^{p} |\tilde{b}_k(u)|^2 = \frac{1}{B^2} \|\mathbf{b}(u)\|_2^2 \leq 1$ for all $u \in K_1$ (and similarly for $\tilde{t}_k$ on D_2). Consequently, the terms $1 - \sum_{k=1}^{p} |\tilde{b}_k(u)|^2$ are non-negative, ensuring that the auxiliary amplitudes defined below are real numbers.

We construct the quantum states in a Hilbert space of dimension $p+2$ ($\cong \mathbb{C}^{p+2}$), spanned by the orthonormal computational basis $\{|1\rangle, \ldots, |p\rangle, |p+1\rangle, |p+2\rangle\}$:

$$|b(u)\rangle = \sum_{k=1}^{p} \tilde{b}_k(u)|k\rangle + \sqrt{1 - \sum_{k=1}^{p} |\tilde{b}_k(u)|^2}\, |p+1\rangle, \tag{46}$$

$$|t(\mathbf{y})\rangle = \sum_{k=1}^{p} \tilde{t}_k(\mathbf{y})|k\rangle + \sqrt{1 - \sum_{k=1}^{p} |\tilde{t}_k(\mathbf{y})|^2}\, |p+2\rangle. \tag{47}$$

These vectors are explicitly normalized (i.e., $\langle b(u)|b(u)\rangle = 1$). Computing their inner product, and utilizing the orthogonality of the basis states ($\langle i|j\rangle = \delta_{ij}$):

$$\begin{aligned}\langle b(u)|t(\mathbf{y})\rangle &= \sum_{k=1}^{p} \tilde{b}_k(u)\tilde{t}_k(\mathbf{y}) \\ &= \sum_{k=1}^{p} \frac{b_k(u)t_k(\mathbf{y})}{BT} = \frac{1}{\lambda}\mathbf{b}(u)\cdot\mathbf{t}(\mathbf{y}).\end{aligned} \tag{48}$$

Note that the cross-terms involving $|p+1\rangle$ and $|p+2\rangle$ vanish because $\langle p+1|p+2\rangle = 0$. Rearranging the terms yields the desired identity: $\mathbf{b}(u)\cdot\mathbf{t}(\mathbf{y}) = \lambda\langle b(u)|t(\mathbf{y})\rangle$. Since $\mathbf{b}(u)$ and $\mathbf{t}(\mathbf{y})$ are continuous and B, T are non-zero constants, the composed mappings remain continuous. ■

C.3 Proof of Theorem 3

We present a constructive proof of the universal approximation property of QuanONet.

Proof

Let $\mathscr{G} : K_1 \to \mathrm{C}(D_2)$ be a continuous operator, with K_1 and D_2 compact. By Theorem 1, for any $\epsilon > 0$ there exist continuous functionals $\hat{b}_k : K_1 \to \mathbb{R}$ and continuous functions $\hat{t}_k : D_2 \to \mathbb{R}$, $k = 1, \ldots, N$, such that

$$\sup_{u\in K_1,\, \mathbf{y}\in D_2} \left| \mathscr{G}(u)(\mathbf{y}) - \sum_{k=1}^{N} \hat{b}_k(u)\hat{t}_k(\mathbf{y}) \right| < \epsilon.$$

Our goal is to construct $\hat{b}_k$ and $\hat{t}_k$ in QuanONet. Since $\hat{b}_k$ and $\hat{t}_k$ are continuous on compact sets, their images are compact and hence bounded. Denote

$$B_{\max} = \max_{1\le k\le N} \sup_{u\in K_1} |\hat{b}_k(u)|, \qquad T_{\max} = \max_{1\le k\le N} \sup_{\mathbf{y}\in D_2} |\hat{t}_k(\mathbf{y})|.$$

Choose an integer $p \ge N+2$ as the dimension of the quantum system. Define scaled amplitudes

$$s_k(\mathbf{y}) = \frac{\hat{t}_k(\mathbf{y})}{\sqrt{p}\, T_{\max}}, \qquad k = 1, \ldots, N,$$

and construct a normalized real vector $\mathbf{t}(\mathbf{y}) \in \mathbb{R}^p$ by

$$\begin{aligned} t_1(\mathbf{y}) &= \frac{1}{\sqrt{p}}, \\ t_{k+1}(\mathbf{y}) &= s_k(\mathbf{y}), \quad k = 1, \dots, N, \\ t_{N+2}(\mathbf{y}) &= \sqrt{1 - \frac{1}{p} - \sum_{k=1}^{N} s_k(\mathbf{y})^2}, \\ t_k(\mathbf{y}) &= 0, \quad k > N+2. \end{aligned}$$

Then $\|\mathbf{t}(\mathbf{y})\|_2 = 1$ for every $\mathbf{y} \in D_2$, and we set $\mathbf{D}(\mathbf{y}) = \mathbf{t}(\mathbf{y})\mathbf{t}(\mathbf{y})^\top$.

Let $\lambda > 0$ and consider the real symmetric arrowhead matrix $\mathbf{M}_0$ defined by

$$(\mathbf{M}_0)_{ii} = 0, \qquad (\mathbf{M}_0)_{1,k+1} = (\mathbf{M}_0)_{k+1,1} = \frac{\lambda}{\sqrt{p-2}} =: r \ \ (k = 1, \dots, N),$$

while $(\mathbf{M}_0)_{1,N+2} = (\mathbf{M}_0)_{N+2,1} = 0$ and all other entries are zero. One checks that $\mathbf{M}_0$ is a normal matrix with eigenvalues $\{\lambda, -\lambda, 0, \dots, 0\}$. Hence there exists a unitary U_0 such that $\mathbf{M}_0 = U_0 \operatorname{diag}(\lambda, -\lambda, 0, \dots, 0) U_0^\dagger$.

Introduce the diagonal unitary $\Phi(u) = \operatorname{diag}\big(1, e^{i\phi_1(u)}, \dots, e^{i\phi_N(u)}, 1, \dots, 1\big)$ with phases $\phi_k(u) \in \mathbb{R}$, and define the modulated observable $\mathbf{M}(u) = \Phi(u)\mathbf{M}_0\Phi(u)^\dagger$. Its nonzero off-diagonal entries are

$$M_{1,k+1}(u) = re^{-i\phi_k(u)}, \qquad M_{k+1,1}(u) = re^{i\phi_k(u)}, \qquad k = 1, \dots, N.$$

Now compute the quantum output:

$$\begin{aligned} \mathscr{Q}(u)(\mathbf{y}) = \langle \mathbf{M}(u), \mathbf{D}(\mathbf{y}) \rangle_F &= \sum_{k=1}^{N} \big(M_{1,k+1}(u) D_{k+1,1}(\mathbf{y}) + M_{k+1,1}(u) D_{1,k+1}(\mathbf{y}) \big) \\ &= \sum_{k=1}^{N} 2r \cos(\phi_k(u))\, t_1(\mathbf{y})\, t_{k+1}(\mathbf{y}) \\ &= \frac{1}{p\, T_{\max}} \sum_{k=1}^{N} 2r \cos(\phi_k(u))\, \hat{t}_k(\mathbf{y}), \end{aligned}$$

where we used $t_1(\mathbf{y}) = 1/\sqrt{p}$ and $t_{k+1}(\mathbf{y}) = s_k(\mathbf{y})$. Notice that the normalization component $t_{N+2}(\mathbf{y})$ does not appear because its coupling in $\mathbf{M}_0$ is zero.

Set $c_k(u) = \dfrac{2r\cos(\phi_k(u))}{p\, T_{\max}}$, so that $\mathscr{Q}(u)(\mathbf{y}) = \sum_{k=1}^{N} c_k(u) \hat{t}_k(\mathbf{y})$. The coefficient $c_k(u)$ satisfies $|c_k(u)| \le \dfrac{2r}{p\, T_{\max}} = \dfrac{2\lambda}{p\, T_{\max}\sqrt{p-2}}$. Choose λ large enough that

$$\frac{2\lambda}{p\, T_{\max}\sqrt{p-2}} \ge B_{\max}, \qquad \text{i.e.} \qquad \lambda \ge \frac{p\, T_{\max}\sqrt{p-2}}{2} B_{\max}.$$

With such a λ, the numbers

$$\tilde{b}_k(u) = \frac{\hat{b}_k(u)\, p\, T_{\max}\sqrt{p-2}}{2\lambda}$$

belong to $[-1, 1]$. Define $\varphi_k(u) = \arccos(\tilde{b}_k(u))$; each φ_k is a continuous function on K_1.

Since the Branch network can represent arbitrary continuous functions on K_1, we may take $\phi_k(u) = \varphi_k(u)$. Then

$$c_k(u) = \frac{2r\cos(\varphi_k(u))}{p\,T_{\max}} = \hat{b}_k(u),$$

and consequently

$$\mathscr{Q}(u)(\mathbf{y}) = \sum_{k=1}^{N} \hat{b}_k(u)\hat{t}_k(\mathbf{y}).$$

Thus, for any $\epsilon > 0$, by choosing p sufficiently large and λ satisfying the inequality above, QuanONet approximates $\mathscr{G}$ with error less than ϵ. This completes the proof. ■

Appendix D. Experimental Setup for High-Dimensional Scaling

In this section, we detail the specific hyperparameter configurations used for the scaling experiments discussed in Section 5.3.1, where the latent dimension p is swept from 4 to 256. The parameters for TF-QuanONet across different latent dimensions p are listed in Table 13, and the corresponding settings for the classical DeepONet baseline are provided in Table 14.

Table 13: Test Relative Error (%) and Parameter Count of TF-QuanONet.

Dim. p	h_b \ h_t	10	20	30	40	50	60	100	150	200	300
4	50	$4.524 \pm 1.073\%$(961)	$3.805 \pm 0.751\%$(1121)	$3.018 \pm 0.160\%$(1281)	$3.105 \pm 0.169\%$(1441)	$3.072 \pm 0.124\%$(1601)	$3.091 \pm 0.251\%$(1761)	$2.800 \pm 0.381\%$(2401)	$3.330 \pm 0.558\%$(3201)	$3.309 \pm 0.633\%$(4001)	$3.377 \pm 0.418\%$(5601)
	100	$3.280 \pm 0.069\%$(1761)	$3.137 \pm 0.252\%$(1921)	$2.842 \pm 0.391\%$(2081)	$2.944 \pm 0.401\%$(2241)	$2.673 \pm 0.199\%$(2401)	$2.509 \pm 0.455\%$(2561)	$2.235 \pm 0.203\%$(3201)	$2.375 \pm 0.242\%$(4001)	$2.833 \pm 0.722\%$(4801)	$2.500 \pm 0.132\%$(6401)
	150	$3.602 \pm 0.196\%$(2561)	$3.187 \pm 0.125\%$(2721)	$2.959 \pm 0.215\%$(2881)	$2.639 \pm 0.441\%$(3041)	$2.642 \pm 0.549\%$(3201)	$2.490 \pm 0.246\%$(3361)	$2.478 \pm 0.503\%$(4001)	$2.374 \pm 0.573\%$(4801)	$2.705 \pm 0.304\%$(5601)	$2.979 \pm 0.112\%$(7201)
	200	$3.551 \pm 0.337\%$(3361)	$3.052 \pm 0.138\%$(3521)	$2.981 \pm 0.501\%$(3681)	$2.607 \pm 0.421\%$(3841)	$2.617 \pm 0.411\%$(4001)	$2.516 \pm 0.328\%$(4161)	$2.512 \pm 0.316\%$(4801)	$2.860 \pm 0.635\%$(5601)	$2.447 \pm 0.238\%$(6401)	$2.449 \pm 0.377\%$(8001)
8	100	-	$2.245 \pm 0.006\%$(2881)	-	$2.293 \pm 0.458\%$(3361)	$1.840 \pm 0.136\%$(3601)	-	$1.663 \pm 0.224\%$(4801)	$1.603 \pm 0.126\%$(6001)	$1.871 \pm 0.445\%$(7201)	$2.036 \pm 0.199\%$(9601)
	200	-	$2.153 \pm 0.169\%$(5281)	-	$1.918 \pm 0.121\%$(5761)	$1.764 \pm 0.138\%$(6001)	-	$1.770 \pm 0.352\%$(7201)	$1.583 \pm 0.290\%$(8401)	$1.518 \pm 0.200\%$(9601)	$1.590 \pm 0.480\%$(12001)
16	100	-	-	-	-	$1.464 \pm 0.040\%$(4801)	-	$1.430 \pm 0.077\%$(6401)	-	-	-
	200	-	-	-	-	$1.498 \pm 0.151\%$(8001)	-	$1.374 \pm 0.317\%$(9601)	-	-	-
32	100	-	-	-	-	$1.400 \pm 0.123\%$(6001)	-	$1.280 \pm 0.087\%$(8001)	-	-	-
64	100	-	-	-	-	$1.323 \pm 0.061\%$(7201)	-	$1.115 \pm 0.065\%$(9601)	-	-	-
128	100	-	-	-	-	$1.216 \pm 0.139\%$(8401)	-	$1.163 \pm 0.132\%$(11201)	-	-	-
256	100	-	-	-	-	$1.169 \pm 0.097\%$(9601)	-	$1.102 \pm 0.122\%$(12801)	-	-	-

Table 14: Test Relative Error (%) and Parameter Count of DeepONet.

Dim. p	B&T Width \ B&T Depth	2	3	4	5
4	4	$9.840 \pm 2.173\%$(453)	$7.211 \pm 2.117\%$(493)	$6.982 \pm 0.435\%$(533)	$8.946 \pm 2.015\%$(573)
	8	$6.222 \pm 0.311\%$(897)	$5.769 \pm 0.155\%$(1041)	$5.443 \pm 0.499\%$(1185)	$5.137 \pm 0.878\%$(1329)
	16	$5.704 \pm 0.168\%$(1785)	$5.557 \pm 0.108\%$(2329)	$5.648 \pm 0.090\%$(2873)	$5.141 \pm 1.029\%$(3417)
	32	$5.740 \pm 0.151\%$(3561)	$5.608 \pm 0.081\%$(5673)	$5.580 \pm 0.134\%$(7785)	$5.056 \pm 1.109\%$(9897)
	64	$5.888 \pm 0.116\%$(7113)	$5.778 \pm 0.139\%$(15433)	$5.892 \pm 0.158\%$(23753)	$6.096 \pm 0.127\%$(32073)
	128	$6.028 \pm 0.212\%$(14217)	$5.905 \pm 0.091\%$(47241)	$6.075 \pm 0.119\%$(80265)	$6.521 \pm 0.224\%$(113289)
	256	$6.066 \pm 0.182\%$(28425)	$5.969 \pm 0.252\%$(160009)	$5.886 \pm 0.149\%$(291593)	$6.242 \pm 0.276\%$(423177)
	512	$6.012 \pm 0.086\%$(56841)	$5.512 \pm 0.100\%$(582153)	$5.629 \pm 0.108\%$(1107465)	$5.915 \pm 0.262\%$(1632777)
	1024	$6.041 \pm 0.266\%$(113673)	$5.499 \pm 0.111\%$(2212873)	$5.599 \pm 0.092\%$(4312073)	$5.703 \pm 0.054\%$(6411273)
8	8	$5.808 \pm 0.155\%$(969)	$4.081 \pm 1.081\%$(1113)	$2.934 \pm 0.432\%$(1257)	$3.240 \pm 0.187\%$(1401)
	16	$4.001 \pm 1.548\%$(1921)	$2.527 \pm 0.643\%$(2465)	$2.492 \pm 0.544\%$(3009)	$1.881 \pm 0.173\%$(3553)
	32	$3.570 \pm 1.263\%$(3825)	$2.404 \pm 0.542\%$(5937)	$1.763 \pm 0.161\%$(8049)	$1.501 \pm 0.295\%$(10161)
	64	$3.902 \pm 1.115\%$(7633)	$1.924 \pm 0.110\%$(15953)	$1.557 \pm 0.191\%$(24273)	$1.447 \pm 0.194\%$(32593)
	128	$3.433 \pm 0.363\%$(15249)	$1.697 \pm 0.137\%$(48273)	$1.534 \pm 0.151\%$(81297)	$1.556 \pm 0.130\%$(114321)
	256	$3.415 \pm 0.213\%$(30481)	$1.700 \pm 0.134\%$(162065)	$1.540 \pm 0.089\%$(293649)	$1.302 \pm 0.283\%$(425233)
	512	$3.523 \pm 0.316\%$(60945)	$1.551 \pm 0.104\%$(586257)	$1.462 \pm 0.279\%$(1111569)	$1.379 \pm 0.250\%$(1636881)
	1024	$4.836 \pm 1.532\%$(121873)	$3.537 \pm 2.186\%$(2221073)	$1.535 \pm 0.128\%$(4320273)	$1.404 \pm 0.223\%$(6419473)
16	16	$3.480 \pm 0.856\%$(2193)	$3.136 \pm 0.267\%$(2737)	$2.024 \pm 0.194\%$(3281)	$1.734 \pm 0.293\%$(3825)
	32	$3.636 \pm 1.546\%$(4353)	$2.138 \pm 0.362\%$(6465)	$1.605 \pm 0.225\%$(8577)	$1.634 \pm 0.258\%$(10689)
	64	$3.107 \pm 0.079\%$(8673)	$1.696 \pm 0.206\%$(16993)	$1.646 \pm 0.128\%$(25313)	$1.363 \pm 0.165\%$(33633)
	128	$3.229 \pm 0.198\%$(17313)	$1.796 \pm 0.204\%$(50337)	$1.612 \pm 0.081\%$(83361)	$1.379 \pm 0.245\%$(116385)
	256	$3.308 \pm 0.266\%$(34593)	$1.701 \pm 0.279\%$(166177)	$1.521 \pm 0.083\%$(297761)	$1.322 \pm 0.307\%$(429345)
	512	$3.474 \pm 0.225\%$(69153)	$3.546 \pm 2.577\%$(594465)	$1.556 \pm 0.372\%$(1119777)	$1.513 \pm 0.234\%$(1645089)
	1024	$3.337 \pm 0.238\%$(138273)	$2.119 \pm 0.798\%$(2237473)	$1.946 \pm 0.421\%$(4336673)	$1.561 \pm 0.056\%$(6435873)
32	32	$3.017 \pm 0.145\%$(5409)	$1.951 \pm 0.447\%$(7521)	$1.574 \pm 0.283\%$(9633)	$1.419 \pm 0.265\%$(11745)
	64	$3.085 \pm 0.306\%$(10753)	$1.672 \pm 0.039\%$(19073)	$1.722 \pm 0.168\%$(27393)	$1.331 \pm 0.101\%$(35713)
	128	$3.207 \pm 0.071\%$(21441)	$1.763 \pm 0.169\%$(54465)	$1.433 \pm 0.258\%$(87489)	$1.268 \pm 0.149\%$(120513)
	256	$3.227 \pm 0.114\%$(42817)	$1.704 \pm 0.218\%$(174401)	$1.637 \pm 0.226\%$(305985)	$1.457 \pm 0.224\%$(437569)
	512	$3.335 \pm 0.313\%$(85569)	$2.879 \pm 0.667\%$(610881)	$2.079 \pm 1.073\%$(1136193)	$1.480 \pm 0.285\%$(1661505)
	1024	$3.075 \pm 0.457\%$(171073)	$2.708 \pm 0.700\%$(2270273)	$1.819 \pm 0.828\%$(4369473)	$1.443 \pm 0.439\%$(6468673)
64	64	$2.769 \pm 0.575\%$(14913)	$1.789 \pm 0.086\%$(23233)	$1.640 \pm 0.327\%$(31553)	$1.292 \pm 0.226\%$(39873)
	128	$3.050 \pm 0.381\%$(29697)	$1.654 \pm 0.075\%$(62721)	$1.505 \pm 0.225\%$(95745)	$1.402 \pm 0.264\%$(128769)
	256	$3.028 \pm 0.171\%$(59265)	$1.666 \pm 0.158\%$(190849)	$1.398 \pm 0.292\%$(322433)	$1.697 \pm 0.295\%$(454017)
	512	$3.672 \pm 1.109\%$(118401)	$2.351 \pm 0.669\%$(643713)	$1.943 \pm 0.668\%$(1169025)	$1.649 \pm 0.235\%$(1694337)
	1024	$2.434 \pm 0.591\%$(236673)	$2.019 \pm 0.479\%$(2335873)	$1.596 \pm 0.206\%$(4435073)	$1.475 \pm 0.205\%$(6534273)
128	128	$2.921 \pm 0.339\%$(46209)	$1.597 \pm 0.078\%$(79233)	$1.467 \pm 0.375\%$(112257)	$1.354 \pm 0.258\%$(145281)
	256	$3.039 \pm 0.213\%$(92161)	$1.879 \pm 0.711\%$(223745)	$1.701 \pm 0.213\%$(355329)	$1.640 \pm 0.284\%$(486913)
	512	$3.139 \pm 0.132\%$(184065)	$1.578 \pm 0.169\%$(709377)	$1.636 \pm 0.417\%$(1234689)	$1.681 \pm 0.298\%$(1760001)
	1024	$1.840 \pm 0.126\%$(367873)	$1.618 \pm 0.151\%$(2467073)	$1.463 \pm 0.175\%$(4566273)	$1.692 \pm 0.506\%$(6665473)
256	256	$3.004 \pm 0.093\%$(157953)	$1.894 \pm 0.732\%$(289537)	$2.461 \pm 1.016\%$(421121)	$2.333 \pm 0.950\%$(552705)
	512	$2.684 \pm 0.421\%$(315393)	$1.694 \pm 0.207\%$(840705)	$1.679 \pm 0.439\%$(1366017)	$1.596 \pm 0.285\%$(1891329)
	1024	$1.841 \pm 0.339\%$(630273)	$1.816 \pm 0.292\%$(2729473)	$1.602 \pm 0.324\%$(4828673)	$1.290 \pm 0.323\%$(6927873)

References

Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021. doi: 10.1038/s43588-021-00084-1.

Ferran Alet, Adarsh Keshav Jeewajee, Maria Bauza Villalonga, Alberto Rodriguez, Tomas Lozano-Perez, and Leslie Kaelbling. Graph element networks: adaptive, structured computation and memory. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 212–222. PMLR, 2019.

Juan Miguel Arrazola, Timjan Kalajdzievski, Christian Weedbrook, and Seth Lloyd. Quantum algorithm for nonhomogeneous linear partial differential equations. *Physical Review A*, 100(3): 032306, 2019.

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.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=vSix3HPYKSU`. Spotlight paper; also available as arXiv:2202.03376.

Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J Coles. Variational quantum linear solver. *arXiv preprint arXiv:1909.05820*, 2019.

Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical review letters*, 87(16):167902, 2001.

Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J. Coles. Generalization in quantum machine learning from few training data. *Nature Communications*, 13(1):4919, 2022. doi: 10.1038/s41467-022-32550-3.

M. Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022. doi: 10.1038/s43588-022-00311-3.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6(4):911–917, 1995.

Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2402–2411. PMLR, 2020.

Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, Shan You, and Dacheng Tao. Learnability of quantum neural networks. *PRX quantum*, 2(4):040337, 2021.

Vedran Dunjko and Hans J Briegel. Machine learning and artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, jun 2018.

Lukas Gonon and Antoine Jacquier. Universal approximation theorem and error bounds for quantum neural networks and quantum reservoirs. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

Junyan He, Shashank Kushwaha, Jaewan Park, Seid Koric, Diab Abueidda, and Iwona Jasiuk. Sequential deep operator networks (s-deeponet) for predicting full-field solutions under time-dependent loads. *Engineering Applications of Artificial Intelligence*, 127:107258, 2024.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

Ben Jaderberg, Antonio A Gentile, Youssef Achari Berrada, Elvira Shishenina, and Vincent E Elfving. Let quantum neural networks choose their own frequencies. *Physical Review A*, 109 (4):042421, 2024.

Nishant Jain, Jonas Landman, Natansh Mathur, and Iordanis Kerenidis. Quantum fourier networks for solving parametric pdes. *Quantum Science and Technology*, 9(3):035026, 2024.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Jennifer Glick, Diego Emilio Hansen, Travis SL Scholten, Paul Nation, Lev S Bishop, et al. Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810*, 2024.

Shi Jin and Nana Liu. Quantum algorithms for computing observables of nonlinear partial differential equations. *arXiv preprint arXiv:2202.07834*, 2022.

Shi Jin, Nana Liu, and Yue Yu. Quantum simulation of partial differential equations via schrödingerization. *Physical Review Letters*, 133(23):230602, 2024.

Shi Jin, Nana Liu, and Yue Yu. Schrodingerization based quantum algorithms for the time-fractional heat equation. *arXiv preprint arXiv:2509.17713*, 2025.

Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.

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.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

Oleksandr Kyriienko, Annie E Paine, and Vincent E Elfving. Solving nonlinear differential equations with differentiable quantum circuits. *Physical Review A*, 103(5):052416, 2021.

Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Shijun Liao. Ham-schrödingerisation: a generic framework of quantum simulation for any nonlinear pdes. *arXiv preprint arXiv:2406.15821*, 2024.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

Jin-Peng Liu, Dong An, Di Fang, Jiasu Wang, Guang Hao Low, and Stephen Jordan. Efficient quantum algorithm for nonlinear reaction–diffusion equations and energy estimation. *Communications in Mathematical Physics*, 404(2):963–1020, 2023.

Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations. *arXiv preprint arXiv:2011.06571*, 2020.

L Lu, P Jin, and GE Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.0319*, 2019.

L Lu, X Meng, Z Mao, E George, and K DeepXDE. A deep learning library for solving differential equations, 2020.

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.

Michael Lubasch, Jaewoo Joo, Pierre Moinier, Martin Kiffner, and Dieter Jaksch. Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1):010301, 2020.

Ashley Montanaro and Sam Pallister. Quantum algorithms and the finite element method. *Physical Review A*, 93(3):032324, 2016.

Furkan Oz, Omer San, and Kursat Kara. An efficient quantum partial differential equation solver with chebyshev points. *Scientific Reports*, 13(1):7767, 2023.

John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. doi: 10.22331/q-2018-08-06-79.

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 E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

Abhijat Sarma, Thomas W Watts, Mudassir Moosa, Yilian Liu, and Peter L McMahon. Quantum variational solving of nonlinear and multidimensional partial differential equations. *Physical Review A*, 109(6):062616, 2024.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

Ruocheng Wang, Zhuo Xia, Ge Yan, and Junchi Yan. Quanonet: Quantum neural operator with application to differential equation. In *International Conference on Machine Learning*, 2025.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. Real-time high-resolution co2 geological storage prediction using nested fourier neural operators. *Energy & Environmental Science*, 16(4):1732–1741, 2023.

Pengpeng Xiao, Muqing Zheng, Anran Jiao, Xiu Yang, and Lu Lu. Quantum deeponet: Neural operators accelerated by quantum computing. *arXiv preprint arXiv:2409.15683*, 2024a.

Y Xiao, LM Yang, C Shu, SC Chew, BC Khoo, YD Cui, and YY Liu. Physics-infoxrmed quantum neural network for solving forward and inverse problems of partial differential equations. *Physics of Fluids*, 36(9), 2024b.

Xusheng Xu, Jiangyu Cui, Zidong Cui, Runhong He, Qingyu Li, Xiaowei Li, Yanling Lin, Jiale Liu, Wuxin Liu, Jiale Lu, et al. Mindspore quantum: a user-friendly, high-performance, and ai-compatible quantum computing framework. *arXiv preprint arXiv:2406.17248*, 2024.

Zhan Yu, Qiuhao Chen, Yuling Jiao, Yinan Li, Xiliang Lu, Xin Wang, and Jerry Zhijian Yang. Non-asymptotic approximation error bounds of parameterized quantum circuits. In *Neural Information Processing Systems*, 2024.

Dan-Bo Zhang, Shi-Liang Zhu, and Z. D. Wang. Protocol for implementing quantum nonparametric learning with trapped ions. *Phys. Rev. Lett.*, 124:010506, Jan 2020.